Amazon CloudTrail

User Guide

Version 1.0



Amazon CloudTrail: User Guide

Table of Contents

What Is Amazon CloudTrail? 1
How CloudTrail works 1
Event history 2
CloudTrail trails 2
CloudTrail channels 3
CloudTrail workflow 3
Event history 3
Trails 4
CloudTrail concepts 5
What are CloudTrail events? 6
What is CloudTrail event history? 10
What are trails? 10
What are organization trails? 10
How do you manage CloudTrail? 11
How do you control access to CloudTrail? 12
How do you log management and data events? 12
How do you log CloudTrail Insights events? 12
How do you perform monitoring with CloudTrail? 12
How does CloudTrail behave regionally and globally? 13
Global service events 14
How does CloudTrail relate to other Amazon monitoring services? 15
Partner solutions 15
CloudTrail supported Regions 15
CloudTrail log file examples 18
CloudTrail log file name format 18
Log file examples 18
CloudTrail supported services and integrations 24
Amazon service integrations with CloudTrail logs 24
CloudTrail integration with Amazon EventBridge 25
CloudTrail integration with Amazon Organizations 26
Amazon service topics for CloudTrail 26
CloudTrail unsupported services 40
Quotas in Amazon CloudTrail 40
CloudTrail tutorial 44
Prerequisites 44
Sign up for an Amazon Web Services account 44
Secure IAM users 44
Grant permissions to use the CloudTrail console 45
Step 1: Review Amazon account activity in event history 45
Step 2: Create your first trail 45
Step 3: View your log files 47
Step 4: Plan for next steps 48
Working with CloudTrail 50
Viewing events with CloudTrail Event history 50
Viewing CloudTrail events in the CloudTrail console 51
Viewing CloudTrail events with the Amazon CLI 55
Viewing CloudTrail Insights events 62
Viewing CloudTrail Insights events in the CloudTrail console 62
Viewing CloudTrail Insights events with the Amazon CLI 69
Creating a trail for your Amazon Web Services account 77
Creating and updating a trail with the console 78
Creating, updating, and managing trails with the Amazon Command Line Interface 97
Creating a trail for an organization 125
Event history and organization trails 127

Best practices for moving from member account trails to organization trails 127
Prepare for creating a trail for your organization 127
Creating a trail for your organization in the console 130
Creating a trail for an organization with the Amazon Command Line Interface 135
Copying trail events to CloudTrail Lake 139
Considerations for copying trail events 140
Required permissions for copying trail events 140
Copy trail events to CloudTrail Lake using the CloudTrail console 143
Getting and viewing your CloudTrail log files 144
Finding your CloudTrail log files 145
Downloading your CloudTrail log files 146
Configuring Amazon SNS notifications for CloudTrail 147
Configuring CloudTrail to send notifications 147
Controlling user permissions for CloudTrail 148
Tips for managing trails 149
Managing CloudTrail costs 149
CloudTrail naming requirements 151
Amazon S3 bucket naming requirements 151
Amazon KMS alias naming requirements 151
Using Amazon CloudTrail with interface VPC endpoints 152
Availability 152
Create a VPC endpoint for CloudTrail 153
Shared subnets 153
Viewing service-linked channels by using the Amazon CLI 153
Get a CloudTrail service-linked channel 153
List all CloudTrail service-linked channels 154
Amazon service events on service-linked channels 155
Working with CloudTrail Lake 156
CloudTrail Lake supported Regions 157
Create an event data store 159
Create an event data store for CloudTrail events 159
Create an event data store for Amazon Config configuration items 163
Create an event data store for events outside of Amazon 170
Manage event data store lifecycles 172
Stop an event data store from ingesting events 173
Copy trail events to an event data store 173
Considerations for copying trail events 174
Required permissions for copying trail events 175
Copy trail events to an event data store 178
Event copy details 179
Create an integration with an event source outside of Amazon 180
Create an integration with a CloudTrail partner 181
Create a custom integration 183
Additional information about integration partners 185
CloudTrail Lake integrations event schema 186
View Lake dashboards 191
Limitations 192
Prerequisites 192
Choosing a dashboard 192
Filtering a dashboard on a date or time range 193
Viewing the query for a dashboard widget 193
Create or edit a query 193
Query editor tools 195
Example queries 196
Run a query and save query results 200
View query results 201
Get and download saved query results 202

Validate saved query results ... 204
Learning resources ... 213
Managing CloudTrail Lake by using the Amazon CLI ... 214
Create an event data store for CloudTrail data events ... 214
Create an event data store for Amazon Config configuration items ... 215
Create an integration to log events from outside Amazon ... 216
Get an event data store ... 221
List all event data stores in an account ... 222
Update an event data store ... 222
Stop ingestion on an event data store ... 223
Start ingestion on an event data store ... 223
Delete an event data store ... 224
Restore an event data store ... 224
List all channels ... 224
Update a channel ... 224
Delete a channel to delete an integration ... 225
Start a query ... 225
Get metadata about a query ... 225
Get query results ... 226
List all queries on an event data store ... 226
Cancel a running query ... 227
CloudTrail Lake SQL constraints ... 227
Supported functions, condition and join operators ... 228
Advanced, multi-table query support ... 229
Supported schemas for event data stores ... 229
Supported CloudWatch metrics ... 235
CloudTrail log files ... 236
Create multiple trails ... 237
Logging management events ... 238
Management events ... 238
Read and write events ... 239
Logging events with the Amazon Command Line Interface ... 240
Logging events with the Amazon SDKs ... 242
Sending events to Amazon CloudWatch Logs ... 242
Logging data events ... 242
Data events ... 243
Read-only and write-only events ... 253
Logging data events with the Amazon Command Line Interface ... 254
Logging events with the Amazon SDKs ... 267
Sending events to Amazon CloudWatch Logs ... 267
Logging Insights events for trails ... 267
Understanding CloudTrail Insights ... 268
Logging Insights events with the Amazon Web Services Management Console ... 273
Logging Insights events with the Amazon Command Line Interface ... 273
Logging events with the Amazon SDKs ... 274
Sending events to Amazon CloudWatch Logs ... 274
Receiving CloudTrail log files from multiple Regions ... 275
Managing data consistency ... 276
Monitoring CloudTrail log files with Amazon CloudWatch Logs ... 277
Sending events to CloudWatch Logs ... 277
Creating CloudWatch alarms for CloudTrail events: examples ... 282
Stopping CloudTrail from sending events to CloudWatch Logs ... 287
CloudWatch log group and log stream naming for CloudTrail ... 288
Role policy document for CloudTrail to use CloudWatch Logs for monitoring ... 288
Receiving CloudTrail log files from multiple accounts ... 290
Redacting bucket owner account IDs for data events called by other accounts ... 290
Setting bucket policy for multiple accounts ... 291

Turning on CloudTrail in additional accounts ... 292
Sharing CloudTrail log files between Amazon accounts ... 297
Scenario 1: Granting access to the account that generated the log files ... 298
Scenario 2: Granting access to all logs ... 299
Creating a role ... 300
Creating an access policy to grant access to accounts you own ... 302
Creating an access policy to grant access to a third party ... 303
Assuming a role ... 304
Stop sharing CloudTrail log files between Amazon accounts ... 307
Validating CloudTrail log file integrity ... 307
Why use it? ... 307
How it works ... 307
Enabling log file integrity validation for CloudTrail ... 308
Validating CloudTrail log file integrity with the Amazon CLI ... 309
CloudTrail digest file structure ... 314
Custom implementations of CloudTrail log file integrity validation ... 319
Using the CloudTrail Processing Library ... 327
Minimum requirements ... 327
Processing CloudTrail logs ... 327
Advanced topics ... 331
Additional resources ... 335
Settings ... 336
Organization delegated administrator ... 336
Required permissions to assign a delegated administrator ... 337
Add a CloudTrail delegated administrator ... 337
Remove a CloudTrail delegated administrator ... 338
Security ... 339
Data protection ... 339
Identity and Access Management ... 340
Audience ... 341
Authenticating with identities ... 341
Managing access using policies ... 343
How Amazon CloudTrail works with IAM ... 344
Identity-based policy examples ... 350
Resource-based policy examples ... 361
Amazon S3 bucket policy for CloudTrail ... 363
Amazon S3 bucket policy for CloudTrail Lake query results ... 368
Amazon SNS topic policy for CloudTrail ... 369
Troubleshooting ... 374
Using service-linked roles ... 377
Amazon managed policies ... 378
Compliance validation ... 380
Resilience ... 380
Infrastructure security ... 381
Cross-service confused deputy prevention ... 381
Security best practices ... 382
CloudTrail detective security best practices ... 382
CloudTrail preventative security best practices ... 383
Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) ... 386
Enabling log file encryption ... 386
Granting permissions to create a KMS key ... 387
Configure Amazon KMS key policies for CloudTrail ... 387
Updating a resource to use your KMS key ... 394
Enabling and disabling CloudTrail log file encryption with the Amazon CLI ... 396
Log event reference ... 400
CloudTrail record contents ... 402
Record fields for Insights events ... 409

Example sharedEventID 410
Services that support TLS details in CloudTrail 411
CloudTrail userIdentity element 416
Examples 416
Fields 417
Values for Amazon STS APIs with SAML and web identity federation 421
Amazon STS source identity 422
Insights insightDetails element 424
Example `insightDetails` block 429
Non-API events captured by CloudTrail 430
Amazon service events 430
Amazon Web Services Management Console sign-in events 431
Document history 438
Earlier updates 457
Amazon glossary 469

What Is Amazon CloudTrail?

Amazon CloudTrail is an Amazon service that helps you enable operational and risk auditing, governance, and compliance of your Amazon account. Actions taken by a user, role, or an Amazon service are recorded as events in CloudTrail. Events include actions taken in the Amazon Web Services Management Console, Amazon Command Line Interface, and Amazon SDKs and APIs.

CloudTrail is enabled on your Amazon account when you create it. When activity occurs in your Amazon account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to Event history. For an ongoing record of activity and events in your Amazon account, create an event data store (p. 159) or create a trail (p. 78). For more information about CloudTrail pricing, see Amazon CloudTrail Pricing.

Visibility into your Amazon account activity is a key aspect of security and operational best practices. You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your Amazon infrastructure. You can identify who or what took which action, what resources were acted upon, when the event occurred, and other details to help you analyze and respond to activity in your Amazon account. Optionally, you can enable Amazon CloudTrail Insights on a trail to help you identify and respond to unusual activity.

You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of event data stores and trails you create, and control how users view CloudTrail events.

Topics

- How CloudTrail works (p. 1)
- CloudTrail workflow (p. 3)
- CloudTrail concepts (p. 5)
- CloudTrail supported Regions (p. 15)
- CloudTrail log file examples (p. 18)
- CloudTrail supported services and integrations (p. 24)
- Quotas in Amazon CloudTrail (p. 40)

How CloudTrail works

CloudTrail is enabled on your Amazon account when you create it. When activity occurs in your Amazon account, that activity is recorded in a CloudTrail event. You can view the past 90 days of recorded API activity (management events) in an Amazon Web Services Region in the CloudTrail console by going to **Event history**.

For an ongoing record of events in your Amazon Web Services account, create a trail. Trails can log events for CloudTrail management, data, and Insights events.

To get started with CloudTrail, see Getting started with Amazon CloudTrail tutorial (p. 44).

For CloudTrail pricing, see Amazon CloudTrail Pricing. For Amazon S3 and Amazon SNS pricing, see Amazon S3 Pricing and Amazon SNS Pricing.

Topics

- Event history (p. 2)
- CloudTrail trails (p. 2)
- CloudTrail channels (p. 3)

Event history

You can easily view management events in the CloudTrail console by going to **Event history**. The **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded API activity (management events) in an Amazon Web Services Region. You can search events in **Event history** by filtering for events on a single attribute. For more information, see Viewing events with CloudTrail Event history.

CloudTrail trails

You can also create a CloudTrail trail to archive, analyze, and respond to changes in your Amazon resources. Trails can log CloudTrail management events, data events, and Insights events.

A *trail* is a configuration that enables delivery of events to an Amazon S3 bucket that you specify. You can also deliver and analyze events in a trail with Amazon CloudWatch Logs and Amazon EventBridge. You can create trails with the CloudTrail console, the Amazon CLI, or the CloudTrail API.

You can create two types of trails for an Amazon account:

A trail that applies to all Regions

When you create a trail that applies to all Regions, CloudTrail records events in each Region and delivers the CloudTrail event log files to an S3 bucket that you specify. If a Region is added after you create a trail that applies to all Regions, that new Region is automatically included, and events in that Region are logged. Creating a multi-Region trail is a recommended best practice since you capture activity in all Regions in your account. All trails you create using the CloudTrail console are multi-Region. You can update a single-Region trail to log all Regions by using the Amazon CLI. For more information, see Creating a trail in the console (basic event selectors) (p. 78) and Converting a trail that applies to one Region to apply to all Regions (p. 101).

A trail that applies to one Region

When you create a trail that applies to one Region, CloudTrail records the events in that Region only. It then delivers the CloudTrail event log files to an Amazon S3 bucket that you specify. You can only create a single-Region trail by using the Amazon CLI. If you create additional single trails, you can have those trails deliver CloudTrail event log files to the same Amazon S3 bucket or to separate buckets. This is the default option when you create a trail using the Amazon CLI or the CloudTrail API. For more information, see Creating, updating, and managing trails with the Amazon Command Line Interface (p. 97).

Note
For both types of trails, you can specify an Amazon S3 bucket from any Region.

Beginning on April 12, 2019, trails are viewable only in the Amazon Regions where they log events. If you create a trail that logs events in all Amazon Regions, it will appear in the console in all Amazon Regions. If you create a trail that only logs events in a single Amazon Region, you can view and manage it only in that Amazon Region.

If you have created an organization in Amazon Organizations, you can create an *organization trail* that logs all events for all Amazon accounts in that organization. Organization trails can apply to all Amazon Regions, or the current Region. Organization trails must be created in the management account or delegated administrator account, and when specified as applying to an organization, are automatically applied to all member accounts in the organization. Member accounts can see the organization trail , but cannot modify or delete it. By default, member accounts do not have access to the log files for an organization trail in the Amazon S3 bucket.

You can change the configuration of a trail after you create it, including whether it logs events in one Region or all Regions. To change a single-Region trail to an all-Region trail, or vice-versa, you must

run the Amazon CLI **update-trail (p. 100)** command. You can also change whether it logs data or CloudTrail Insights events. Changing whether a trail logs events in one Region or in all Regions affects which events are logged. For more information, see Managing trails with the Amazon CLI (p. 103) (Amazon CLI), and Working with CloudTrail log files (p. 236).

By default, CloudTrail event log files from trails are encrypted using Amazon S3 server-side encryption (SSE). You can also choose to encrypt your log files with an Amazon Key Management Service (Amazon KMS) key. You can store your log files in your bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.

CloudTrail publishes log files multiple times an hour, about every 5 minutes. These log files contain API calls from services in the account that support CloudTrail. For more information, see CloudTrail supported services and integrations (p. 24).

Note
CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed.
CloudTrail captures actions made directly by the user or on behalf of the user by an Amazon service. For example, an Amazon CloudFormation `CreateStack` call can result in additional API calls to Amazon EC2, Amazon RDS, Amazon EBS, or other services as required by the Amazon CloudFormation template. This behavior is normal and expected. You can identify if the action was taken by an Amazon service with the `invokedby` field in the CloudTrail event.

CloudTrail channels

CloudTrail supports service-linked channels.

Service-linked channels

Amazon services can create a service-linked channel to receive CloudTrail events on your behalf. The Amazon service creating the service-linked channel configures advanced event selectors for the channel and specifies whether the channel applies to all Regions, or a single Region.

Using the Amazon CLI, you can view information about any CloudTrail service-linked channels created by Amazon services. For more information, see Viewing service-linked channels for CloudTrail by using the Amazon CLI (p. 153).

CloudTrail workflow

This section provides information about CloudTrail features and the tasks you can perform for these features.

CloudTrail is enabled by default when you open your Amazon account and you automatically have access to view the last 90 days of management events for your Amazon Web Services account using **Event history**.

For an ongoing record of events in your Amazon Web Services account past 90 days, create a trail. Trails can log events for CloudTrail management, data, and Insights events.

Event history

View event history for your Amazon account

You can view and search the last 90 days of events recorded by CloudTrail in the CloudTrail console or by using the Amazon CLI. For more information, see Viewing events with CloudTrail Event history (p. 50).

Download events

You can download a CSV or JSON file containing up to the past 90 days of CloudTrail events for your Amazon account. For more information, see Downloading events (p. 54).

Trails

Create a trail

A trail enables CloudTrail to deliver log files to your Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the Amazon partition and delivers the log files to the S3 bucket that you specify. For more information, see Creating a trail for your Amazon Web Services account (p. 77).

Log management and data events

Configure your trails to log read-only, write-only, or all management and data events. By default, trails log management events. For more information, see Logging management events (p. 238) and Logging data events (p. 242).

Log Insights events

Configure your trails to log Insights events to help you identify and respond to unusual activity associated with management API calls. For more information, see Logging Insights events for trails (p. 267).

View Insights events

After you enable CloudTrail Insights on a trail, you can view up to 90 days of Insights events by using the CloudTrail console or the Amazon CLI. For more information, see Viewing CloudTrail Insights events (p. 62).

Download Insights events

After you enable CloudTrail Insights on a trail, you can download a CSV or JSON file containing up to the past 90 days of Insights events for your trail. For more information, see Downloading Insights events (p. 69).

Create and subscribe to an Amazon SNS topic

Subscribe to a topic to receive notifications about log file delivery to your bucket. Amazon SNS can notify you in multiple ways, including programmatically with Amazon Simple Queue Service. For information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

> **Note**
> If you want to receive SNS notifications about log file deliveries from all Regions, specify only one SNS topic for your trail. If you want to programmatically process all events, see Using the CloudTrail Processing Library (p. 327).

View your log files

Use Amazon S3 to retrieve log files. For information, see Getting and viewing your CloudTrail log files (p. 144).

Monitor events with CloudWatch Logs

You can configure your trail to send events to CloudWatch Logs. You can then use CloudWatch Logs to monitor your account for specific API calls and events. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

> **Note**
> If you configure a trail that applies to all Regions to send events to a CloudWatch Logs log group, CloudTrail sends events from all Regions to a single log group.

Enable log encryption

Log file encryption provides an extra layer of security for your log files. For more information, see Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386).

Enable log file integrity

Log file integrity validation helps you verify that log files have remained unchanged since CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

Share log files with other Amazon accounts

You can share log files between accounts. For more information, see Sharing CloudTrail log files between Amazon accounts (p. 297).

Aggregate logs from multiple accounts

You can aggregate log files from multiple accounts to a single bucket. For more information, see Receiving CloudTrail log files from multiple accounts (p. 290).

Manage user permissions

Use Amazon Identity and Access Management (IAM) to manage which users have permissions to create, configure, or delete trails; start and stop logging; and access buckets that have log files. For more information, see Granting permissions for CloudTrail administration (p. 355).

Register a delegated administrator to manage your organization's CloudTrail resources

You can register a delegated administrator to manage your organization's CloudTrail trails. For more information, see Organization delegated administrator (p. 336).

Work with partner solutions

Analyze your CloudTrail output with a partner solution that integrates with CloudTrail. Partner solutions offer a broad set of capabilities, such as change tracking, troubleshooting, and security analysis. For more information, see the Amazon CloudTrail partner page.

CloudTrail concepts

This section summarizes basic concepts related to CloudTrail.

Contents

- What are CloudTrail events? (p. 6)
  - What are management events? (p. 6)
  - What are data events? (p. 6)
  - What are Insights events? (p. 9)
- What is CloudTrail event history? (p. 10)
- What are trails? (p. 10)
- What are organization trails? (p. 10)
- How do you manage CloudTrail? (p. 11)
  - CloudTrail console (p. 11)
  - CloudTrail CLI (p. 11)
  - CloudTrail APIs (p. 11)
  - Amazon SDKs (p. 11)
  - Why use tags for trails? (p. 11)
- How do you control access to CloudTrail? (p. 12)
- How do you log management and data events? (p. 12)
- How do you log CloudTrail Insights events? (p. 12)

- How do you perform monitoring with CloudTrail? (p. 12)
    - CloudWatch Logs, EventBridge, and CloudTrail (p. 12)
- How does CloudTrail behave regionally and globally? (p. 13)
    - What are the advantages of applying a trail to all Regions? (p. 13)
    - What happens when you apply a trail to all Regions? (p. 13)
    - Multiple trails per Region (p. 14)
    - Amazon Security Token Service and CloudTrail (p. 14)
- Global service events (p. 14)
- How does CloudTrail relate to other Amazon monitoring services? (p. 15)
- Partner solutions (p. 15)

What are CloudTrail events?

An event in CloudTrail is the record of an activity in an Amazon account. This activity can be an action taken by an IAM identity, or service that is monitorable by CloudTrail. CloudTrail events provide a history of both API and non-API account activity made through the Amazon Web Services Management Console, Amazon SDKs, command line tools, and other Amazon services. There are three types of events that can be logged in CloudTrail: management events, data events, and CloudTrail Insights events. By default, trails log management events, but not data or Insights events.

All event types use a CloudTrail JSON log format.

> **Note**
> CloudTrail does not log all Amazon services and all events. For more information about which APIs are logged for a specific service, see documentation for that service in CloudTrail supported services and integrations (p. 24).

What are management events?

Management events provide information about management operations that are performed on resources in your Amazon account. These are also known as *control plane operations*. Example management events include:

- Configuring security (for example, Amazon Identity and Access Management `AttachRolePolicy` API operations).
- Registering devices (for example, Amazon EC2 `CreateDefaultVpc` API operations).
- Configuring rules for routing data (for example, Amazon EC2 `CreateSubnet` API operations).
- Setting up logging (for example, Amazon CloudTrail `CreateTrail` API operations).

Management events can also include non-API events that occur in your account. For example, when a user signs in to your account, CloudTrail logs the `ConsoleLogin` event. For more information, see Non-API events captured by CloudTrail (p. 430). For a list of management events that CloudTrail logs for Amazon services, see CloudTrail supported services and integrations (p. 24).

What are data events?

Data events provide information about the resource operations performed on or in a resource. These are also known as *data plane operations*. Data events are often high-volume activities.

The following table shows the data event types available for trails. The **Data event type (console)** column shows the appropriate selection in the console. The **resources.Type** column shows the `resources.Type` value that you would specify to include data events of that type in your trail.

For trails, you can use basic or advanced event selectors to log data events for Amazon S3 buckets and bucket objects, Lambda functions, and DynamoDB tables (shown in the first three rows of the table). You can use only advanced event selectors to log the data event types shown in the remaining rows.

Amazon service	Description	Data event type (console)	resources.Type
Amazon DynamoDB	Amazon DynamoDB object-level API activity on tables (for example, `PutItem`, `DeleteItem`, and `UpdateItem` API operations). For more information about DynamoDB events, see DynamoDB data plane events in CloudTrail.	DynamoDB	`AWS::DynamoDB::Table`
Amazon Lambda	Amazon Lambda function execution activity (the `Invoke` API).	Lambda	`AWS::Lambda::Function`
Amazon S3	Amazon S3 object-level API activity (for example, `GetObject`, `DeleteObject`, and `PutObject` API operations) on buckets and objects in buckets.	S3	`AWS::S3::Object`
Amazon CloudTrail	CloudTrail `PutAuditEvents` activity on a CloudTrail Lake channel (p. 180) that is used to log events from outside Amazon.	CloudTrail	`AWS::CloudTrail::Channel`
Amazon CodeWhisperer	Amazon CodeWhisperer API activity on a profile.	CodeWhisperer	`AWS::CodeWhisperer::Profile`
Amazon Cognito	Amazon Cognito API activity on Amazon Cognito identity pools.	Cognito Identity Pools	`AWS::Cognito::IdentityPool`
Amazon DynamoDB	Amazon DynamoDB API activity on streams.	DynamoDB Streams	`AWS::DynamoDB::Stream`
Amazon Elastic Block Store	Amazon Elastic Block Store (EBS) direct APIs, such as `PutSnapshotBlock`, `GetSnapshotBlock`, and `ListChangedBlocks` on Amazon EBS snapshots.	Amazon EBS direct APIs	`AWS::EC2::Snapshot`

Amazon service	Description	Data event type (console)	resources.Type
Amazon EMR	Amazon EMR API activity on a write-ahead log workspace.	EMR write-ahead log workspace	`AWS::EMRWAL::Workspace`
Amazon FinSpace	Amazon FinSpace API activity on environments.	FinSpace	`AWS::FinSpace::Environment`
Amazon Glue	Amazon Glue API activity on tables that were created by Lake Formation. **Note** Amazon Glue data events for tables are currently supported only in the following regions: • US East (N. Virginia) • US East (Ohio) • US West (Oregon) • Europe (Ireland) • Asia Pacific (Tokyo) Region	Lake Formation	`AWS::Glue::Table`
Amazon GuardDuty	Amazon GuardDuty API activity for a detector.	GuardDuty detector	`AWS::GuardDuty::Detector`
Amazon HealthImaging	Amazon HealthImaging API activity on data stores.	Medical Imaging data store	`AWS::MedicalImaging::Datastore`
Amazon Kendra Intelligent Ranking	Amazon Kendra Intelligent Ranking API activity on rescore execution plans.	Kendra Ranking	`AWS::KendraRanking::ExecutionPlan`
Amazon Managed Blockchain	Amazon Managed Blockchain API activity on a network.	Managed Blockchain network	`AWS::ManagedBlockchain::Network`

Amazon service	Description	Data event type (console)	resources.Type
Amazon Managed Blockchain	Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, such as `eth_getBalance` or `eth_getBlockByNumber`.	Managed Blockchain	`AWS::ManagedBlockchain::Node`
Amazon SageMaker	Amazon SageMaker API activity on feature stores.	SageMaker feature store	`AWS::SageMaker::FeatureGroup`
Amazon SageMaker	Amazon SageMaker API activity on experiment trial components.	SageMaker metrics experiment trial component	`AWS::SageMaker::ExperimentTrialComponent`
Amazon S3	Amazon S3 API activity on access points.	S3 Access Point	`AWS::S3::AccessPoint`
Amazon S3	Amazon S3 Object Lambda access points API activity, such as calls to `CompleteMultipartUpload` and `GetObject`.	S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`
Amazon S3 on Outposts	Amazon S3 on Outposts object-level API activity.	S3 Outposts	`AWS::S3Outposts::Object`
Amazon Systems Manager	Amazon Systems Manager API activity on control channels.	Systems Manager	`AWS::SSMMessages::ControlChannel`
Amazon Verified Permissions	Amazon Verified Permissions API activity on a policy store.	Amazon Verified Permissions	`AWS::VerifiedPermissions::PolicyStore`

Data events are not logged by default when you create a trail. To record CloudTrail data events, you must explicitly add the supported resources or resource types for which you want to collect activity. For more information, see Creating a trail (p. 78).

Additional charges apply for logging data events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

What are Insights events?

CloudTrail Insights events capture unusual API call rate or error rate activity in your Amazon account by analyzing CloudTrail management activity. If you have Insights events enabled, and CloudTrail detects unusual activity, Insights events are logged to a different folder or prefix in the destination S3 bucket for your trail. You can also see the type of insight and the incident time period when you view Insights events on the CloudTrail console. Insights events provide relevant information, such as the associated API, error code, incident time, and statistics, that help you understand and act on unusual activity. Unlike other types of events captured in a CloudTrail trail, Insights events are logged only when CloudTrail detects changes in your account's API usage or error rate logging that differ significantly from the account's typical usage patterns. Examples of activity that might generate Insights events include:

- Your account typically logs no more than 20 Amazon S3 `deleteBucket` API calls per minute, but your account starts to log an average of 100 `deleteBucket` API calls per minute. An Insights event is logged at the start of the unusual activity, and another Insights event is logged to mark the end of the unusual activity.
- Your account typically logs 20 calls per minute to the Amazon EC2 `AuthorizeSecurityGroupIngress` API, but your account starts to log zero calls to `AuthorizeSecurityGroupIngress`. An Insights event is logged at the start of the unusual activity, and ten minutes later, when the unusual activity ends, another Insights event is logged to mark the end of the unusual activity.
- Your account typically logs less than one `AccessDeniedException` error in a seven-day period on the Amazon Identity and Access Management API, `DeleteInstanceProfile`. Your account starts to log an average of 12 `AccessDeniedException` errors per minute on the `DeleteInstanceProfile` API call. An Insights event is logged at the start of the unusual error rate activity, and another Insights event is logged to mark the end of the unusual activity.

These examples are provided for illustration purposes only. Your results may vary depending on your use case.

Insights events are disabled by default when you create a trail. To log CloudTrail Insights events, you must explicitly enable Insights event collection on a new or existing trail, and the trail must log CloudTrail management events. For more information, see Creating a trail (p. 78) and Logging Insights events for trails (p. 267).

Additional charges apply for logging CloudTrail Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

What is CloudTrail event history?

CloudTrail event history provides a viewable, searchable, downloadable, and immutable record of the past 90 days of CloudTrail management events in an Amazon Web Services Region. You can use this history to gain visibility into actions taken in your Amazon account in the Amazon Web Services Management Console, Amazon SDKs, command line tools, and other Amazon services. You can customize your view of event history in the CloudTrail console by selecting which columns are displayed. For more information, see Viewing events with CloudTrail Event history (p. 50).

What are trails?

A trail is a configuration that enables delivery of CloudTrail events to an Amazon S3 bucket, CloudWatch Logs, and Amazon EventBridge. You can use a trail to filter the CloudTrail events you want delivered, encrypt your CloudTrail event log files with an Amazon KMS key, and set up Amazon SNS notifications for log file delivery. For more information about how to create and manage a trail, see Creating a trail for your Amazon Web Services account (p. 77).

What are organization trails?

An organization trail is a configuration that enables delivery of CloudTrail events in the management account, delegated administrator account, and all member accounts in an Amazon Organizations organization to the same Amazon S3 bucket, CloudWatch Logs, and Amazon EventBridge. Creating an organization trail helps you define a uniform event logging strategy for your organization.

When you create an organization trail, a trail with the name that you give it will be created in every Amazon account that belongs to your organization. Users with CloudTrail permissions in member accounts will be able to see this trail (including the trail ARN) when they log into the Amazon CloudTrail console from their Amazon accounts, or when they run Amazon CLI commands such as `describe-trails` (although member accounts must use the ARN for the organization trail, and not the name, when using the Amazon CLI). However, users in member accounts will not have sufficient permissions

to delete the organization trail, turn logging on or off, change what types of events are logged, or otherwise alter the organization trail in any way. For more information about Amazon Organizations, see Organizations Terminology and Concepts. For more information about creating and working with organization trails, see Creating a trail for an organization (p. 125).

How do you manage CloudTrail?

CloudTrail console

You can use and manage the service with the Amazon CloudTrail console. The console provides a user interface for performing many CloudTrail tasks such as:

- Viewing recent events and event history for your Amazon account.
- Downloading a filtered or complete file of the last 90 days of events.
- Creating and editing CloudTrail trails.
- Configuring CloudTrail trails, including:
  - Selecting an Amazon S3 bucket for trails.
  - Setting a prefix.
  - Configuring delivery to CloudWatch Logs.
  - Using Amazon KMS keys for encryption of trail data.
  - Enabling Amazon SNS notifications for log file delivery on trails.
  - Adding and managing tags for your trails.

Beginning on April 12, 2019, trails will be viewable only in the Amazon Regions where they log events. If you create a trail that logs events in all Amazon Regions, it will appear in the console in all Amazon Regions. If you create a trail that only logs events in a single Amazon Region, you can view and manage it only in that Amazon Region.

For more information about the Amazon Web Services Management Console, see Amazon Web Services Management Console.

CloudTrail CLI

The Amazon Command Line Interface is a unified tool that you can use to interact with CloudTrail from the command line. For more information, see the Amazon Command Line Interface User Guide. For a complete list of CloudTrail CLI commands, see Available Commands.

CloudTrail APIs

In addition to the console and the CLI, you can also use the CloudTrail RESTful APIs to program CloudTrail directly. For more information, see the Amazon CloudTrail API Reference.

Amazon SDKs

As an alternative to using the CloudTrail API, you can use one of the Amazon SDKs. Each SDK consists of libraries and sample code for various programming languages and platforms. The SDKs provide a convenient way to create programmatic access to CloudTrail. For example, you can use the SDKs to sign requests cryptographically, manage errors, and retry requests automatically. For more information, see the Tools for Amazon Web Services page.

Why use tags for trails?

A tag is a customer-defined key and optional value that can be assigned to Amazon resources, such as CloudTrail trails, Amazon S3 buckets used to store CloudTrail log files, Amazon Organizations

organizations and organizational units, and many more. By adding the same tags to trails and to the Amazon S3 buckets you use to store log files for trails, you can make it easier to manage, search for, and filter these resources with Amazon Resource Groups. You can implement tagging strategies to help you consistently, effectively, and easily find and manage your resources. For more information, see Amazon Tagging Strategies.

How do you control access to CloudTrail?

Amazon Identity and Access Management is a web service that enables Amazon Web Services (Amazon) customers to securely control access to Amazon resources. Using IAM, you can centrally manage permissions that control which CloudTrail resources users can access. For more information about controlling user permissions, see Controlling user permissions for CloudTrail (p. 148).

How do you log management and data events?

By default, trails log management events for your Amazon account and don't include data events. You can choose to create or update trails to log data events. Only events that match your trail settings are delivered to your Amazon S3 bucket, and optionally to an Amazon CloudWatch Logs log group. If the event doesn't match the settings for a trail, the trail doesn't log the event. For more information, see Working with CloudTrail log files (p. 236).

How do you log CloudTrail Insights events?

Amazon CloudTrail Insights helps Amazon users identify and respond to unusual volumes of API calls or errors logged on API calls by continuously analyzing CloudTrail management events. An Insights event is a record of unusual levels of `write` management API activity, or unusual levels of errors returned on management API activity. The details page of an Insights event shows the event as a graph of unusual activity, and shows the start and end times of the unusual activity, along with the baseline that is used to determine whether the activity is unusual. By default, trails don't log CloudTrail Insights events. In the console, you can choose to log Insights events when you create or update a trail. When you use the CloudTrail API, you can log Insights events by editing the settings of an existing trail with the **PutInsightSelectors** API. Additional charges apply for logging CloudTrail Insights events. For more information, see Logging Insights events for trails (p. 267) and Amazon CloudTrail Pricing.

How do you perform monitoring with CloudTrail?

CloudWatch Logs, EventBridge, and CloudTrail

Amazon CloudWatch is a web service that collects and tracks metrics to monitor your Amazon Web Services (Amazon) resources and the applications that you run on Amazon. Amazon CloudWatch Logs is a feature of CloudWatch that you can use specifically to monitor log data. Integration with CloudWatch Logs enables CloudTrail to send events containing API activity in your Amazon account to a CloudWatch Logs log group. CloudTrail events that are sent to CloudWatch Logs can trigger alarms according to the metric filters you define. You can optionally configure CloudWatch alarms to send notifications or make changes to the resources that you are monitoring based on log stream events that your metric filters extract. Using CloudWatch Logs, you can also track CloudTrail events alongside events from the operating system, applications, or other Amazon services that are sent to CloudWatch Logs. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

Amazon EventBridge is an Amazon service that delivers a near real-time stream of system events that describe changes in Amazon resources. In EventBridge, you can create rules that respond to events recorded by CloudTrail. For more information, see Create a rule in Amazon EventBridge.

You can deliver events that you are subscribed to on your trail to EventBridge. When you create a rule with the EventBridge console, choose either the `Amazon API Call via CloudTrail` detail-type to

deliver CloudTrail data and management events, or the `Amazon Insight via CloudTrail` detail-type to deliver Insights events.

To record events with a detail-type value of `Amazon API Call via CloudTrail`, you must have an active trail that is logging management or data events. For more information about how to create a trail, see Creating a trail (p. 78).

To record events with a detail-type value of `Amazon Insight via CloudTrail`, you must have an active trail that is logging Insights events. For information about logging Insights events, see Logging Insights events for trails (p. 267).

How does CloudTrail behave regionally and globally?

A trail can be applied to all Regions or a single Region. As a best practice, create a trail that applies to all Regions in the Amazon partition in which you are working. This is the default setting when you create a trail in the CloudTrail console.

> **Note**
> *Turning on a trail* means that you create a trail and start delivery of CloudTrail event log files to an Amazon S3 bucket. In the CloudTrail console, logging is turned on automatically when you create a trail.

What are the advantages of applying a trail to all Regions?

A trail that applies to all Amazon Regions has the following advantages:

- The configuration settings for the trail apply consistently across all Amazon Regions.
- You receive CloudTrail events from all Amazon Regions in a single Amazon S3 bucket and, optionally, in a CloudWatch Logs log group.
- You manage trail configuration for all Amazon Regions from one location.
- You immediately receive events from a new Amazon Region. When a new Amazon Region is launched, CloudTrail automatically creates a copy of all of your Region trails for you in the new Region with the same settings as your original trail.
- You don't need to create trails in Amazon Regions that you don't use often in order to monitor for unusual activity. Any activity in any Amazon Region is logged in a trail that applies to all Amazon Regions.

What happens when you apply a trail to all Regions?

When you apply a trail to all Amazon Regions, CloudTrail uses the trail that you create in a particular Region to create trails with identical configurations in all other Regions in your account.

This has the following effects:

- CloudTrail delivers log files for account activity from all Amazon Regions to the single Amazon S3 bucket that you specify, and, optionally, to a CloudWatch Logs log group.
- If you configured an Amazon SNS topic for the trail, SNS notifications about log file deliveries in all Amazon Regions are sent to that single SNS topic.
- If you enabled it, log file integrity validation is enabled for the trail in all Amazon Regions. For information, see Validating CloudTrail log file integrity (p. 307).

Regardless of whether a trail is multi-Region or single-Region, events sent to Amazon EventBridge are received in each Region's event bus, rather than in one single event bus.

Multiple trails per Region

If you have different but related user groups, such as developers, security personnel, and IT auditors, you can create multiple trails per Region. This allows each group to receive its own copy of the log files.

CloudTrail supports five trails per Region. A trail that applies to all Amazon Regions counts as one trail in every Region.

The following example is a Region with five trails:

- You create two trails in the US West (N. California) Region that apply to this Region only.
- You create two more trails in US West (N. California) Region that apply to all Amazon Regions.
- You create a trail in the Asia Pacific (Sydney) Region that applies to all Amazon Regions. This trail also exists as a trail in the US West (N. California) Region.

Trails appear in the Amazon Region where they exist. Trails that log events in all Amazon Regions appear in every Region. You can view a list of trails in an Amazon Region in the **Trails** page of the CloudTrail console. For more information, see Updating a trail (p. 89). For CloudTrail pricing, see Amazon CloudTrail Pricing.

Amazon Security Token Service and CloudTrail

Amazon Security Token Service (Amazon STS) is a service that has a global endpoint and also supports Region-specific endpoints. An endpoint is a URL that is the entry point for web service requests. For example, `https://cloudtrail.us-west-2.amazonaws.com` is the US West (Oregon) regional entry point for the Amazon CloudTrail service. Regional endpoints help reduce latency in your applications.

When you use an Amazon STS Region-specific endpoint, the trail in that Region delivers only the Amazon STS events that occur in that Region. For example, if you are using the endpoint `sts.us-west-2.amazonaws.com`, the trail in us-west-2 delivers only the Amazon STS events that originate from us-west-2. For more information about Amazon STS regional endpoints, see Activating and Deactivating Amazon STS in an Amazon Region in the *IAM User Guide*.

For a complete list of Amazon regional endpoints, see Amazon Regions and Endpoints in the *Amazon Web Services General Reference*. For details about events from the global Amazon STS endpoint, see Global service events (p. 14).

Global service events

Important
As of November 22, 2021, Amazon CloudTrail makes Amazon CloudFront events available only in the Region where the event was processed, China (Ningxia) Region, cn-northwest-1.
For trails monitoring global service events, be sure to convert *single-Region trails* in China (Beijing) Region, cn-north-1, to *multi-Region trails*, to include events from China (Ningxia) Region, cn-northwest-1. For more information about capturing CloudFront events, see Enabling and disabling global service event logging (p. 102) later in this section.

For most services, events are recorded in the Region where the action occurred. For global services such as Amazon Identity and Access Management (IAM), Amazon STS, and Amazon CloudFront, events are delivered to any trail that includes global services.

For most global services, events are logged as occurring in the China (Beijing) Region, but some global service events are logged as occurring in the China (Ningxia) Region.

To avoid receiving duplicate global service events, remember the following:

- Global service events are delivered by default to trails that are created using the CloudTrail console. Events are delivered to the bucket for the trail.

- If you have multiple single Region trails, consider configuring your trails so that global service events are delivered in only one of the trails. For more information, see Enabling and disabling global service event logging (p. 102).
- If you change the configuration of a trail from logging all Regions to logging a single Region, global service event logging is turned off automatically for that trail. Similarly, if you change the configuration of a trail from logging a single Region to logging all Regions, global service event logging is turned on automatically for that trail.

  For more information about changing global service event logging for a trail, see Enabling and disabling global service event logging (p. 102).

Example:

1. You create a trail in the CloudTrail console. By default, this trail logs global service events.
2. You have multiple single Region trails.
3. You do not need to include global services for the single Region trails. Global service events are delivered for the first trail. For more information, see Creating, updating, and managing trails with the Amazon Command Line Interface (p. 97).

   Note
   When you create or update a trail with the Amazon CLI, Amazon SDKs, or CloudTrail API, you can specify whether to include or exclude global service events for trails. You cannot configure global service event logging from the CloudTrail console.

How does CloudTrail relate to other Amazon monitoring services?

CloudTrail adds another dimension to the monitoring capabilities already offered by Amazon. It does not change or replace logging features you might already be using, such as those for Amazon S3 or Amazon CloudFront subscriptions. Amazon CloudWatch focuses on performance monitoring and system health. CloudTrail focuses on API activity. Although CloudTrail does not report on system performance or health, you can use CloudTrail with CloudWatch alarms to notify you about activity that you might be interested in.

Partner solutions

Amazon partners with third-party specialists in logging and analysis to provide solutions that use CloudTrail output. For more information, visit the CloudTrail detail page at Amazon CloudTrail.

CloudTrail supported Regions

Note
For information about Regions supported by CloudTrail Lake, see CloudTrail Lake supported Regions (p. 157).

Region Name	Region	Endpoint	Protocol	Amazon Account ID	Support Date
US East (N. Virginia)	us-east-1	cloudtrail.us-east-1.amazonaws.com	HTTPS	086441151436	11/13/2013

Region Name	Region	Endpoint	Protocol	Amazon Account ID	Support Date
US East (Ohio)	us-east-2	cloudtrail.us-east-2.amazonaws.com	HTTPS	475085895292	10/17/2016
US West (N. California)	us-west-1	cloudtrail.us-west-1.amazonaws.com	HTTPS	388731089494	05/13/2014
US West (Oregon)	us-west-2	cloudtrail.us-west-2.amazonaws.com	HTTPS	113285607260	11/13/2013
Canada (Central)	ca-central-1	cloudtrail.ca-central-1.amazonaws.com	HTTPS	819402241893	12/08/2016
Africa (Cape Town)	af-south-1	cloudtrail.af-south-1.amazonaws.com	HTTPS	525921808201	04/22/2020
Asia Pacific (Hong Kong)	ap-east-1	cloudtrail.ap-east-1.amazonaws.com	HTTPS	119688915426	04/24/2019
Asia Pacific (Mumbai)	ap-south-1	cloudtrail.ap-south-1.amazonaws.com	HTTPS	977081816279	06/27/2016
Asia Pacific (Hyderabad)	ap-south-2	cloudtrail.ap-south-2.amazonaws.com	HTTPS	582488909970	11/22/2022
Asia Pacific (Tokyo)	ap-northeast-1	cloudtrail.ap-northeast-1.amazonaws.com	HTTPS	216624486486	06/30/2014
Asia Pacific (Seoul)	ap-northeast-2	cloudtrail.ap-northeast-2.amazonaws.com	HTTPS	492519147666	01/06/2016
Asia Pacific (Osaka)	ap-northeast-3	cloudtrail.ap-northeast-3.amazonaws.com	HTTPS	765225791966	02/12/2018
Asia Pacific (Singapore)	ap-southeast-1	cloudtrail.ap-southeast-1.amazonaws.com	HTTPS	903692715234	06/30/2014
Asia Pacific (Sydney)	ap-southeast-2	cloudtrail.ap-southeast-2.amazonaws.com	HTTPS	284668455005	05/13/2014
Asia Pacific (Jakarta)	ap-southeast-3	cloudtrail.ap-southeast-3.amazonaws.com	HTTPS	069019280451	12/13/2021
Asia Pacific (Melbourne)	ap-southeast-4	cloudtrail.ap-southeast-4.amazonaws.com	HTTPS	187074758985	01/23/2023

Region Name	Region	Endpoint	Protocol	Amazon Account ID	Support Date
China (Beijing)	cn-north-1	cloudtrail.cn-north-1.amazonaws.com.cn	HTTPS	193415116832	03/01/2014
China (Ningxia)	cn-northwest-1	cloudtrail.cn-northwest-1.amazonaws.com.cn	HTTPS	681348832753	12/11/2017
Europe (Frankfurt)	eu-central-1	cloudtrail.eu-central-1.amazonaws.com	HTTPS	035351147821	10/23/2014
Europe (Zurich)	eu-central-2	cloudtrail.eu-central-2.amazonaws.com	HTTPS	453052556044	11/09/2022
Europe (Stockholm)	eu-north-1	cloudtrail.eu-north-1.amazonaws.com	HTTPS	829690693026	12/11/2018
Europe (Ireland)	eu-west-1	cloudtrail.eu-west-1.amazonaws.com	HTTPS	859597730677	05/13/2014
Europe (London)	eu-west-2	cloudtrail.eu-west-2.amazonaws.com	HTTPS	282025262664	12/13/2016
Europe (Paris)	eu-west-3	cloudtrail.eu-west-3.amazonaws.com	HTTPS	262312530599	12/18/2017
Europe (Milan)	eu-south-1	cloudtrail.eu-south-1.amazonaws.com	HTTPS	669305197877	04/27/2020
Europe (Spain)	eu-south-2	cloudtrail.eu-south-2.amazonaws.com	HTTPS	757211635381	11/16/2022
Israel (Tel Aviv)	il-central-1	cloudtrail.il-central-1.amazonaws.com	HTTPS	683224464357	07/31/2023
Middle East (UAE)	me-central-1	cloudtrail.me-central-1.amazonaws.com	HTTPS	585772288577	08/30/2022
Middle East (Bahrain)	me-south-1	cloudtrail.me-south-1.amazonaws.com	HTTPS	034638983726	07/29/2019
Amazon GovCloud (US-East)	us-gov-east-1	cloudtrail.us-gov-east-1.amazonaws.com	HTTPS	886388586500	11/12/2018
Amazon GovCloud (US-West)	us-gov-west-1	cloudtrail.us-gov-west-1.amazonaws.com	HTTPS	608710470296	08/16/2011
South America (São Paulo)	sa-east-1	cloudtrail.sa-east-1.amazonaws.com	HTTPS	814480443879	06/30/2014

For more information about using CloudTrail in the Amazon GovCloud (US-East) Region, see Amazon GovCloud (US-East) Endpoints in the *Amazon GovCloud (US) User Guide*.

For information about CloudTrail in the China (Beijing) Region, see China (Beijing) Region Endpoints in the *Amazon Web Services General Reference* and Amazon CloudTrail in the *Getting Started with Amazon Amazon Guide*.

CloudTrail log file examples

CloudTrail monitors events for your account. If you create a trail, it delivers those events as log files to your Amazon S3 bucket. If you create an event data store in CloudTrail Lake, events are logged to your event data store. Event data stores do not use S3 buckets.

Topics

- CloudTrail log file name format (p. 18)
- Log file examples (p. 18)

CloudTrail log file name format

CloudTrail uses the following file name format for the log file objects that it delivers to your Amazon S3 bucket:

```
AccountID_CloudTrail_RegionName_YYYYMMDDTHHmmZ_UniqueString.FileNameFormat
```

- The YYYY, MM, DD, HH, and mm are the digits of the year, month, day, hour, and minute when the log file was delivered. Hours are in 24-hour format. The Z indicates that the time is in UTC.

  Note
  A log file delivered at a specific time can contain records written at any point before that time.
- The 16-character `UniqueString` component of the log file name is there to prevent overwriting of files. It has no meaning, and log processing software should ignore it.
- `FileNameFormat` is the encoding of the file. Currently, this is `json.gz`, which is a JSON text file in compressed gzip format.

Example CloudTrail Log File Name

```
111122223333_CloudTrail_us-east-2_20150801T0210Z_Mu0KsOhtH1ar15ZZ.json.gz
```

Log file examples

A log file contains one or more records. The following examples are snippets of logs that show the records for an action that started the creation of a log file.

Contents

- Amazon EC2 log examples (p. 18)
- IAM log examples (p. 20)
- Error code and message log example (p. 22)
- CloudTrail Insights event log example (p. 23)

Amazon EC2 log examples

Amazon Elastic Compute Cloud (Amazon EC2) provides resizeable computing capacity in the Amazon Web Services Cloud. You can launch virtual servers, configure security and networking, and manage

storage. Amazon EC2 can also scale up or down quickly to handle changes in requirements or spikes in popularity, thereby reducing your need to forecast server traffic. For more information, see the Amazon EC2 User Guide for Linux Instances.

The following example shows that an IAM user named Alice used the Amazon CLI to call the Amazon EC2 `StartInstances` action using the `ec2-start-instances` command for instance `i-ebeaf9e2`.


```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21:22:54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools 1.6.12.2",
    "requestParameters": {"instancesSet": {"items": [{"instanceId": "i-ebeaf9e2"}]}},
    "responseElements": {"instancesSet": {"items": [{
        "instanceId": "i-ebeaf9e2",
        "currentState": {
            "code": 0,
            "name": "pending"
        },
        "previousState": {
            "code": 80,
            "name": "stopped"
        }
    }]}}
}]}
```


The following example shows that an IAM user named Alice used the Amazon CLI to call the Amazon EC2 `StopInstances`action by using the `ec2-stop-instances`.


```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21:01:59Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StopInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools 1.6.12.2",
    "requestParameters": {
        "instancesSet": {"items": [{"instanceId": "i-ebeaf9e2"}]},
        "force": false
    },
    "responseElements": {"instancesSet": {"items": [{
        "instanceId": "i-ebeaf9e2",
        "currentState": {
```

```
            "code": 64,
            "name": "stopping"
        },
        "previousState": {
            "code": 16,
            "name": "running"
        }
    }]}}
}]}
```

The following example shows that the Amazon EC2 console backend called the `CreateKeyPair` action in response to requests initiated by the IAM user Alice. Note that the `responseElements` contain a hash of the key pair and that Amazon removed the key material.

```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice",
        "sessionContext": {"attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2014-03-06T15:15:06Z"
        }}
    },
    "eventTime": "2014-03-06T17:10:34Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateKeyPair",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "72.21.198.64",
    "userAgent": "EC2ConsoleBackend, aws-sdk-java/Linux/x.xx.fleetxen Java_HotSpot(TM)_64-
Bit_Server_VM/xx",
    "requestParameters": {"keyName": "mykeypair"},
    "responseElements": {
        "keyName": "mykeypair",
        "keyFingerprint": "30:1d:46:d0:5b:ad:7e:1b:b6:70:62:8b:ff:38:b5:e9:ab:5d:b8:21",
        "keyMaterial": "\u003csensitiveDataRemoved\u003e"
    }
}]}
```

IAM log examples

Amazon Identity and Access Management (IAM) is a web service that helps you securely control access to Amazon resources. With IAM, you can centrally manage permissions that control which Amazon resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources. For more information, see the IAM User Guide.

The following example shows that the IAM user Alice used the Amazon CLI to call the `CreateUser` action to create a new user named Bob.

```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
```

```
        "userName": "Alice"
    },
    "eventTime": "2014-03-24T21:11:59Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/7",
    "requestParameters": {"userName": "Bob"},
    "responseElements": {"user": {
        "createDate": "Mar 24, 2014 9:11:59 PM",
        "userName": "Bob",
        "arn": "arn:aws:iam::123456789012:user/Bob",
        "path": "/",
        "userId": "EXAMPLEUSERID"
    }}
}]}
```

The following example shows that the IAM user Alice used the Amazon Web Services Management Console to call the `AddUserToGroup` action to add Bob to the administrator group.

```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice",
        "sessionContext": {"attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2014-03-25T18:45:11Z"
        }}
    },
    "eventTime": "2014-03-25T21:08:14Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "AddUserToGroup",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "AWSConsole",
    "requestParameters": {
        "userName": "Bob",
        "groupName": "admin"
    },
    "responseElements": null
}]}
```

The following example shows that the IAM user Alice used the Amazon CLI to call the `CreateRole` action to create a new IAM role.

```
{
    "Records": [{
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
        "eventTime": "2014-03-25T20:17:37Z",
```

```
        "eventSource": "iam.amazonaws.com",
        "eventName": "CreateRole",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/7",
        "requestParameters": {
            "assumeRolePolicyDocument": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement
\": [\n    {\n      \"Sid\": \"\",
             \n\"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\":
 \"arn:aws:iam::210987654321:root\"\n      },\n      \"Action\": \"sts:AssumeRole\"\n
  }\n  ]\n}",
            "roleName": "TestRole"
        },
        "responseElements": {
            "role": {
                "assumeRolePolicyDocument": "%7B%0A%20%20%22Version%22%3A
%20%222012-10-17%22%2C%0A%20%20%22Statement%22%3A%20%5B%0A%20%20%20%20%7B%0A
%20%20%20%20%20%20%22Sid%22%3A%20%22%22%2C%0A%20%20%20%20%20%20%22Effect%22%3A%20%22Allow
%22%2C%0A%20%20%20%20%20%20%22Principal%22%3A%20%7B%0A%20%20%20%20%20%20%20%20%22AWS
%22%3A%20%22arn%3Aaws%3Aiam%3A%3A803981987763%3Aroot%22%0A%20%20%20%20%20%20%7D%2C%0A
%20%20%20%20%20%20%22Action%22%3A%20%22sts%3AAssumeRole%22%0A%20%20%20%20%7D%0A%20%20%5D%0A
%7D",
                "roleName": "TestRole",
                "roleId": "AROAIUU2EOWSWPGX2UJUO",
                "arn": "arn:aws:iam::123456789012:role/TestRole",
                "createDate": "Mar 25, 2014 8:17:37 PM",
                "path": "/"
            }
        }
    }]
}
```

Error code and message log example

The following example shows that the IAM user Alice used the Amazon CLI to call the `UpdateTrail` action to update a trail named `myTrail2`, but the trail name was not found. The log shows this error in the `errorCode` and `errorMessage` elements.

```
{"Records": [{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2016-07-14T19:15:45Z",
    "eventSource": "cloudtrail.amazonaws.com",
    "eventName": "UpdateTrail",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.182",
    "userAgent": "aws-cli/1.10.32 Python/2.7.9 Windows/7 botocore/1.4.22",
    "errorCode": "TrailNotFoundException",
    "errorMessage": "Unknown trail: myTrail2 for the user: 123456789012",
    "requestParameters": {"name": "myTrail2"},
    "responseElements": null,
    "requestID": "5d40662a-49f7-11e6-97e4-d9cb6ff7d6a3",
    "eventID": "b7d4398e-b2f0-4faa-9c76-e2d316a8d67f",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}]}
```

CloudTrail Insights event log example

The following example shows a CloudTrail Insights event log. An Insights event is actually a pair of events that mark the start and end of a period of unusual write management API activity or error response activity. The `state` field shows whether the event was logged at the start or end of the period of unusual activity. The event name, `UpdateInstanceInformation`, is the same name as the Amazon Systems Manager API for which CloudTrail analyzed management events to determine that unusual activity occurred. Although the start and end events have unique `eventID` values, they also have a `sharedEventID` value that is used by the pair. The Insights event shows the `baseline`, or the normal pattern of activity, the `insight`, or average unusual activity that triggered the start Insights event, and in the end event, the `insight` value for the average unusual activity over the duration of the Insights event. For more information about CloudTrail Insights, see Logging Insights events for trails (p. 267).


```
{
    "Records": [
        {
            "eventVersion": "1.07",
            "eventTime": "2019-11-14T00:51:00Z",
            "awsRegion": "us-east-1",
            "eventID": "EXAMPLE8-9621-4d00-b913-beca2EXAMPLE",
            "eventType": "AwsCloudTrailInsight",
            "recipientAccountId": "123456789012",
            "sharedEventID": "EXAMPLE2-1729-42f1-b735-5d8c0EXAMPLE",
            "insightDetails": {
                "state": "Start",
                "eventSource": "ssm.amazonaws.com",
                "eventName": "UpdateInstanceInformation",
                "insightType": "ApiCallRateInsight",
                "insightContext": {
                    "statistics": {
                        "baseline": {
                            "average": 85.4202380952
                        },
                        "insight": {
                            "average": 664
                        }
                    }
                }
            },
            "eventCategory": "Insight"
        },
        {
            "eventVersion": "1.07",
            "eventTime": "2019-11-14T00:52:00Z",
            "awsRegion": "us-east-1",
            "eventID": "EXAMPLEc-28be-486c-8928-49ce6EXAMPLE",
            "eventType": "AwsCloudTrailInsight",
            "recipientAccountId": "123456789012",
            "sharedEventID": "EXAMPLE2-1729-42f1-b735-5d8c0EXAMPLE",
            "insightDetails": {
                "state": "End",
                "eventSource": "ssm.amazonaws.com",
                "eventName": "UpdateInstanceInformation",
                "insightType": "ApiCallRateInsight",
                "insightContext": {
                    "statistics": {
                        "baseline": {
                            "average": 85.4202380952
                        },
                        "insight": {
                            "average": 664
                        },
                        "insightDuration": 1
```

```
                    }
                }
            },
            "eventCategory": "Insight"
        }
    ]
}
```

CloudTrail supported services and integrations

CloudTrail supports logging events for many Amazon services. You can find the specifics for each supported service in that service's guide. Links to those service-specific topics are provided below. In addition, some Amazon services can be used to analyze and act upon data collected in CloudTrail logs. You can browse an overview of those service integrations here.

> **Note**
> To see the list of supported Regions for each service, see Service endpoints and quotas in the *Amazon Web Services General Reference*.

Topics

- Amazon service integrations with CloudTrail logs (p. 24)
- CloudTrail integration with Amazon EventBridge (p. 25)
- CloudTrail integration with Amazon Organizations (p. 26)
- Amazon service topics for CloudTrail (p. 26)
- CloudTrail unsupported services (p. 40)

Amazon service integrations with CloudTrail logs

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

Amazon Service	Topic	Description
Amazon Athena	Querying Amazon CloudTrail Logs	Using Athena with CloudTrail logs is a powerful way to enhance your analysis of Amazon service activity. For example, you can use queries to identify trends and further isolate activity by attribute, such as source IP address or user. You can automatically create tables for querying logs directly from the CloudTrail console, and use those tables to run queries in Athena. For more information, see Creating a Table for CloudTrail Logs in the CloudTrail Console in the Amazon Athena User Guide.

Amazon Service	Topic	Description
		Note Running queries in Amazon Athena incurs additional costs. For more information, see Amazon Athena Pricing.
Amazon CloudWatch Logs	Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277)	You can configure CloudTrail with CloudWatch Logs to monitor your trail logs and be notified when specific activity occurs. For example, you can define CloudWatch Logs metric filters that will trigger CloudWatch alarms and send notifications to you when those alarms are triggered. **Note** Standard pricing for Amazon CloudWatch and Amazon CloudWatch Logs applies. For more information, see Amazon CloudWatch Pricing.

CloudTrail integration with Amazon EventBridge

Amazon EventBridge is an Amazon service that delivers a near real-time stream of system events that describe changes in Amazon resources. In EventBridge, you can create rules that responds to events recorded by CloudTrail. For more information, see Create a rule in Amazon EventBridge.

You can deliver events that you are subscribed to on your trail to EventBridge. When you create a rule with the EventBridge console, choose either the `Amazon API Call via CloudTrail` detail-type to deliver CloudTrail data and management events, or the `Amazon Insight via CloudTrail` detail-type to deliver Insights events.

To record events with a detail-type value of `Amazon API Call via CloudTrail`, you must have an active trail that is logging management or data events. For more information about how to create a trail, see Creating a trail (p. 78).

To record events with a detail-type value of `Amazon Insight via CloudTrail`, you must have an active trail that is logging Insights events. For information about logging Insights events, see Logging Insights events for trails (p. 267).

Note
Events from API actions that start with the keywords `List`, `Get`, or `Describe` are not processed by EventBridge, with the exception of events from the following STS actions: `GetFederationToken` and `GetSessionToken`.

CloudTrail integration with Amazon Organizations

The management account for an Amazon Organizations organization can configure a delegated administrator to manage the organization's CloudTrail resources. You can create a trail in the management account or delegated administrator account for an organization that collects all event data for all Amazon accounts in an organization in Amazon Organizations. This is called an organization trail. Creating an organization trail helps you define a uniform event logging strategy for your organization. An organization trail is applied automatically to each Amazon account in your organization. Users in member accounts can see these trails but cannot modify them, and by default cannot see the log files created for the organization trail. For more information, see Creating a trail for an organization (p. 125).

Amazon service topics for CloudTrail

You can learn more about how the events for individual Amazon services are recorded in CloudTrail logs, including example events for that service in log files. For more information about how specific Amazon services integrate with CloudTrail, see the topic about integration in the individual guide for that service.

Services that are still in preview, or not yet released for general availability (GA), or which don't have public APIs, are not considered supported. CloudTrail does not currently log Amazon VPC endpoint policy-specific events.

> **Note**
> To see the list of supported Regions for each service, see Service endpoints and quotas in the *Amazon Web Services General Reference*.
> For information about which services log data events, see Data events (p. 243).

Amazon Service	CloudTrail Topics	Support began
Alexa for Business	Logging Alexa for Business Administration Calls Using Amazon CloudTrail	11/29/2017
Amazon API Gateway	Log API management calls to Amazon API Gateway Using Amazon CloudTrail	07/09/2015
Amazon AppFlow	Logging Amazon AppFlow API calls with Amazon CloudTrail	04/22/2020
Amazon AppStream 2.0	Logging Amazon AppStream 2.0 API Calls with Amazon CloudTrail	04/25/2019
Amazon Athena	Logging Amazon Athena API Calls with Amazon CloudTrail	05/19/2017
Amazon Aurora	Monitoring Amazon Aurora API calls in Amazon CloudTrail	08/31/2018
Amazon Braket	Amazon Braket API logging with CloudTrail	08/12/2020
Amazon Chime	Log Amazon Chime Administration Calls Using Amazon CloudTrail	09/27/2017

Amazon Service	CloudTrail Topics	Support began
Amazon Cloud Directory	Logging Amazon Cloud Directory API Calls Using Amazon CloudTrail	01/26/2017
Amazon CloudFront	Using Amazon CloudTrail to Capture Requests Sent to the CloudFront API	05/28/2014
Amazon CloudSearch	Logging Amazon CloudSearch Configuration Service Calls Using Amazon CloudTrail	10/16/2014
Amazon CloudWatch	Logging Amazon CloudWatch API Calls in Amazon CloudTrail	04/30/2014
Amazon CloudWatch Events	Logging Amazon CloudWatch Events API Calls in Amazon CloudTrail	01/16/2016
Amazon CloudWatch Logs	Logging Amazon CloudWatch Logs API Calls in Amazon CloudTrail	03/10/2016
Amazon CodeCatalyst	Logging CodeCatalyst API calls in connected Amazon Web Services accounts using Amazon CloudTrail	12/01/2022
Amazon CodeGuru Reviewer	Logging Amazon CodeGuru Reviewer API Calls with Amazon CloudTrail	12/02/2019
Amazon CodeWhisperer	Amazon CloudTrail and CodeWhisperer APIs	04/13/2023
Amazon Cognito	Logging Amazon Cognito API Calls with Amazon CloudTrail	02/18/2016
Amazon Comprehend	Logging Amazon Comprehend API Calls with Amazon CloudTrail	01/17/2018
Amazon Comprehend Medical	Logging Amazon Comprehend Medical API Calls by Using Amazon CloudTrail	11/27/2018
Amazon Connect	Logging Amazon Connect API Calls with Amazon CloudTrail	12/11/2019
Amazon Data Lifecycle Manager	Logging Amazon Data Lifecycle Manager API Calls Using Amazon CloudTrail	07/24/2018
Amazon Detective	Logging Amazon Detective API calls with Amazon CloudTrail	03/31/2020
Amazon DevOps Guru	Logging Amazon DevOps Guru API calls with Amazon CloudTrail	05/04/2021

Amazon Service	CloudTrail Topics	Support began
Amazon DocumentDB (with MongoDB compatibility)	Logging Amazon DocumentDB API Calls with Amazon CloudTrail	01/09/2019
Amazon DynamoDB	Logging DynamoDB Operations By Using Amazon CloudTrail	05/28/2015
Amazon EC2	Logging API Calls Using Amazon CloudTrail	11/13/2013
Amazon EC2 Auto Scaling	Logging Auto Scaling API Calls By Using CloudTrail	07/16/2014
Amazon EC2 Image Builder	Logging EC2 Image Builder API calls using CloudTrail	12/02/2019
Amazon Elastic Block Store (Amazon EBS) EBS direct APIs	Logging API Calls Using Amazon CloudTrail Log API Calls for the EBS direct APIs with Amazon CloudTrail	Amazon EBS: 11/13/2013 EBS direct APIs: 06/30/2020
Amazon Elastic Container Registry (Amazon ECR)	Logging Amazon ECR API Calls By Using Amazon CloudTrail	12/21/2015
Amazon Elastic Container Service (Amazon ECS)	Logging Amazon ECS API Calls By Using Amazon CloudTrail	04/09/2015
Amazon Elastic File System (Amazon EFS)	Logging Amazon EFS API Calls with Amazon CloudTrail	06/28/2016
Amazon Elastic Kubernetes Service (Amazon EKS)	Logging Amazon EKS API Calls with Amazon CloudTrail	06/05/2018
Amazon Elastic Transcoder	Logging Amazon Elastic Transcoder API Calls with Amazon CloudTrail	10/27/2014
Amazon ElastiCache	Logging Amazon ElastiCache API Calls Using Amazon CloudTrail	09/15/2014
Amazon EMR	Logging Amazon EMR API Calls in Amazon CloudTrail	04/04/2014
Amazon EMR on EKS	Logging Amazon EMR on EKS API calls using Amazon CloudTrail	12/09/2020
Amazon EventBridge	EventBridge information in Amazon CloudTrail	07/11/2019
Amazon FinSpace	Querying Amazon CloudTrail logs	10/18/2022
Amazon Forecast	Logging Amazon Forecast API Calls with Amazon CloudTrail	11/28/2018

Amazon Service	CloudTrail Topics	Support began
Amazon Fraud Detector	Logging Amazon Fraud Detector API Calls with Amazon CloudTrail	01/09/2020
Amazon FSx for Lustre	Logging Amazon FSx for Lustre API Calls with Amazon CloudTrail	01/11/2019
Amazon FSx for Windows File Server	Monitoring with Amazon CloudTrail	11/28/2018
Amazon GameLift	Logging Amazon GameLift API Calls with Amazon CloudTrail	01/27/2016
Amazon GameSparks	Log GameSparks API calls with Amazon CloudTrail	03/23/2022
Amazon GuardDuty	Logging Amazon GuardDuty API Calls with Amazon CloudTrail	02/12/2018
Amazon Honeycode	Logging Amazon Honeycode API Calls with Amazon CloudTrail	06/24/2020
Amazon Inspector	Logging Amazon Inspector API calls with Amazon CloudTrail	04/20/2016
Amazon Interactive Video Service	Logging Amazon IVS API Calls with Amazon CloudTrail	07/15/2020
Amazon Kendra	Logging Amazon Kendra API calls with Amazon CloudTrail and Logging Amazon Kendra Intelligent Ranking API calls with Amazon CloudTrail logs	05/11/2020
Amazon Keyspaces (for Apache Cassandra)	Logging Amazon Keyspaces API calls with Amazon CloudTrail	01/13/2020
Amazon Kinesis Data Analytics	Monitoring Amazon Kinesis Data Analytics with Amazon CloudTrail (SQL Applications) and Monitoring Amazon Kinesis Data Analytics with Amazon CloudTrail (Apache Flink Applications)	03/22/2019
Amazon Kinesis Data Firehose	Monitoring Amazon Kinesis Data Firehose API Calls with Amazon CloudTrail	03/17/2016
Amazon Kinesis Data Streams	Logging Amazon Kinesis Data Streams API Calls Using Amazon CloudTrail	04/25/2014
Amazon Kinesis Video Streams	Logging Kinesis Video Streams API Calls with Amazon CloudTrail	05/24/2018

Amazon Service	CloudTrail Topics	Support began
Amazon Lex	Logging Amazon Lex API Calls with CloudTrail	08/15/2017
Amazon Lightsail	Logging Lightsail API Calls with Amazon CloudTrail	12/23/2016
Amazon Location Service	Logging and monitoring with Amazon CloudTrail	12/15/2020
Amazon Lookout for Equipment	Monitoring Amazon Lookout for Equipment calls with Amazon CloudTrail	12/01/2020
Amazon Lookout for Metrics	Viewing Amazon Lookout for Metrics API activity in Amazon CloudTrail	12/08/2020
Amazon Lookout for Vision	Logging Amazon Lookout for Vision calls with Amazon CloudTrail	12/01/2020
Amazon Machine Learning	Logging Amazon ML API Calls By Using Amazon CloudTrail	12/10/2015
Amazon Macie	Log Amazon Macie API calls using Amazon CloudTrail	05/13/2020
Amazon Managed Blockchain	Logging Amazon Managed Blockchain API calls using Amazon CloudTrail Logging Ethereum for Managed Blockchain API calls using Amazon CloudTrail (Preview)	04/01/2019
Amazon Managed Grafana	Logging Amazon Managed Grafana API calls using Amazon CloudTrail	12/15/2020
Amazon Managed Service for Prometheus	Logging Amazon Managed Service for Prometheus API calls using Amazon CloudTrail	12/15/2020
Amazon Managed Streaming for Apache Kafka	Logging Amazon MSK API Calls with Amazon CloudTrail	12/11/2018
Amazon Managed Workflows for Apache Airflow	Monitoring Amazon MWAA API activity with Amazon CloudTrail	11/24/2020
Amazon MemoryDB for Redis	Logging Amazon MemoryDB for Redis API calls with Amazon CloudTrail	08/19/2021
Amazon MQ	Logging Amazon MQ API Calls Using Amazon CloudTrail	07/19/2018
Amazon Neptune	Logging Amazon Neptune API Calls Using Amazon CloudTrail	05/30/2018

Amazon Service	CloudTrail Topics	Support began
Amazon Nimble Studio	Logging Nimble Studio calls using Amazon CloudTrail	06/19/2023
Amazon OpenSearch Service	Auditing Amazon OpenSearch Service Domains with Amazon CloudTrail	10/01/2015
Amazon Personalize	Logging Amazon Personalize API Calls with Amazon CloudTrail	11/28/2018
Amazon Pinpoint	Logging Amazon Pinpoint API Calls with Amazon CloudTrail	02/06/2018
Amazon Pinpoint SMS and Voice API	Logging Amazon Pinpoint API Calls with Amazon CloudTrail	11/16/2018
Amazon Polly	Logging Amazon Polly API Calls with Amazon CloudTrail	11/30/2016
Amazon Quantum Ledger Database (Amazon QLDB)	Logging Amazon QLDB API Calls with Amazon CloudTrail	09/10/2019
Amazon QuickSight	Logging Operations with CloudTrail	04/28/2017
Amazon Relational Database Service (Amazon RDS)	Logging Amazon RDS API Calls Using Amazon CloudTrail	11/13/2013
Amazon RDS Performance Insights	Logging Amazon RDS API Calls Using Amazon CloudTrail The Amazon RDS Performance Insights API is a subset of the Amazon RDS API.	06/21/2018
Amazon Redshift	Logging Amazon Redshift API Calls with Amazon CloudTrail	06/10/2014
Amazon Rekognition	Logging Amazon Rekognition API Calls Using Amazon CloudTrail	04/6/2018
Amazon Route 53	Using Amazon CloudTrail to Capture Requests Sent to the Route 53 API	02/11/2015
Amazon Route 53 Application Recovery Controller	Logging Amazon Route 53 Application Recovery Controller API calls using Amazon CloudTrail	07/27/2021
Amazon S3	Logging Amazon S3 API Calls By Using Amazon CloudTrail	Management events: 09/01/2015 Data events: 11/21/2016
Amazon S3 Glacier	Logging S3 Glacier API Calls By Using Amazon CloudTrail	12/11/2014

Amazon Service	CloudTrail Topics	Support began
Amazon SageMaker	Logging Amazon SageMaker API Calls with Amazon CloudTrail	01/11/2018
Amazon Security Lake	Logging Amazon Security Lake API calls using CloudTrail	05/30/2023
Amazon Simple Email Service (Amazon SES)	Logging Amazon SES API Calls By Using Amazon CloudTrail	05/07/2015
Amazon Simple Notification Service (Amazon SNS)	Logging Amazon Simple Notification Service API Calls By Using Amazon CloudTrail	10/09/2014
Amazon Simple Queue Service (Amazon SQS)	Logging Amazon SQS API Actions Using Amazon CloudTrail	07/16/2014
Amazon Simple Workflow Service (Amazon SWF)	Logging Amazon Simple Workflow Service API Calls with Amazon CloudTrail	05/13/2014
Amazon Textract	Logging Amazon Textract API Calls with Amazon CloudTrail	05/29/2019
Amazon Timestream	Logging Timestream API calls with Amazon CloudTrail	09/30/2020
Amazon Transcribe	Logging Amazon Transcribe API Calls with Amazon CloudTrail	06/28/2018
Amazon Translate	Logging Amazon Translate API Calls with Amazon CloudTrail	04/04/2018
Amazon Verified Permissions	Logging Amazon Verified Permissions API calls using Amazon CloudTrail	06/13/2023
Amazon Virtual Private Cloud (Amazon VPC)	Logging API Calls Using Amazon CloudTrail The Amazon VPC API is a subset of the Amazon EC2 API.	11/13/2013
Amazon VPC Lattice	CloudTrail logs	03/31/2023
Amazon WorkDocs	Logging Amazon WorkDocs API Calls By Using Amazon CloudTrail	08/27/2014
Amazon WorkMail	Logging Amazon WorkMail API Calls Using Amazon CloudTrail	12/12/2017
Amazon WorkSpaces	Logging Amazon WorkSpaces API Calls by Using CloudTrail	04/09/2015
Amazon WorkSpaces Web	Logging Amazon WorkSpaces Web API calls using Amazon CloudTrail	11/30/2021

Amazon Service	CloudTrail Topics	Support began
Application Auto Scaling	Logging Application Auto Scaling API calls with Amazon CloudTrail	10/31/2016
Amazon Amplify	Logging Amplify API calls using Amazon CloudTrail	11/30/2020
Amazon App Mesh	Logging App Mesh API Calls with Amazon CloudTrail	Amazon App Mesh 10/30/2019 App Mesh Envoy Management Service 03/18/2022
Amazon App Runner	Logging App Runner API calls with Amazon CloudTrail	05/18/2021
Amazon AppFabric	Logging Amazon AppFabric API calls using Amazon CloudTrail	06/27/2023
Amazon Application Cost Profiler	Amazon Application Cost Profiler API Reference	05/13/2021
Amazon Application Discovery Service	Logging Application Discovery Service API Calls with Amazon CloudTrail	05/12/2016
Amazon AppSync	Logging Amazon AppSync API Calls with Amazon CloudTrail	02/13/2018
Amazon Artifact	Logging Amazon Artifact API calls with Amazon CloudTrail	01/27/2023
Amazon Audit Manager	Logging Amazon Audit Manager API calls with Amazon CloudTrail	12/07/2020
Amazon Auto Scaling	Logging Amazon Auto Scaling API Calls By Using CloudTrail	08/15/2018
Amazon Backup	Logging Amazon Backup API Calls with Amazon CloudTrail	02/04/2019
Amazon Batch	Logging Amazon Batch API Calls with Amazon CloudTrail	1/10/2018
Amazon Billing and Cost Management	Logging Amazon Billing and Cost Management API Calls with Amazon CloudTrail	06/07/2018
Amazon BugBust	Logging BugBust API calls using CloudTrail	06/24/2021
Amazon Certificate Manager	Using Amazon CloudTrail	03/25/2016
Amazon Certificate Manager Private Certificate Authority	Using CloudTrail	06/06/2019
Amazon Clean Rooms	Logging Amazon Clean Rooms API calls using Amazon CloudTrail	03/21/2023

Amazon Service	CloudTrail Topics	Support began
Amazon Cloud Map	Logging Amazon Cloud Map API Calls with Amazon CloudTrail	11/28/2018
Amazon Cloud9	Logging Amazon Cloud9 API Calls with Amazon CloudTrail	01/21/2019
Amazon CloudFormation	Logging Amazon CloudFormation API Calls in Amazon CloudTrail	04/02/2014
Amazon CloudHSM	Logging Amazon CloudHSM API Calls By Using Amazon CloudTrail	01/08/2015
Amazon CloudShell	Logging and monitoring in Amazon CloudShell	12/15/2020
Amazon CloudTrail	Amazon CloudTrail API Reference (All CloudTrail API calls are logged by CloudTrail.)	11/13/2013
Amazon CodeArtifact	Logging CodeArtifact API calls with Amazon CloudTrail	06/10/2020
Amazon CodeBuild	Logging Amazon CodeBuild API Calls with Amazon CloudTrail	12/01/2016
Amazon CodeCommit	Logging Amazon CodeCommit API Calls with Amazon CloudTrail	01/11/2017
Amazon CodeDeploy	Monitoring Deployments with Amazon CloudTrail	12/16/2014
Amazon CodePipeline	Logging CodePipeline API Calls By Using Amazon CloudTrail	07/09/2015
AWS CodeStar	Logging AWS CodeStar API Calls with Amazon CloudTrail	06/14/2017
AWS CodeStar Notifications	Logging AWS CodeStar Notifications API Calls with Amazon CloudTrail	11/05/2019
Amazon Config	Logging Amazon Config API Calls By with Amazon CloudTrail	02/10/2015
Amazon Control Tower	Logging Amazon Control Tower Actions with Amazon CloudTrail	08/12/2019
Amazon Data Pipeline	Logging Amazon Data Pipeline API Calls by using Amazon CloudTrail	12/02/2014
Amazon Database Migration Service (Amazon DMS)	Logging Amazon Database Migration Service API Calls Using Amazon CloudTrail	02/04/2016

Amazon Service	CloudTrail Topics	Support began
Amazon DataSync	Logging Amazon DataSync API Calls with Amazon CloudTrail	11/26/2018
Amazon Device Farm	Logging Amazon Device Farm API Calls By Using Amazon CloudTrail	07/13/2015
Amazon Direct Connect	Logging Amazon Direct Connect API Calls in Amazon CloudTrail	03/08/2014
Amazon Directory Service	Logging Amazon Directory Service API Calls by Using CloudTrail	05/14/2015
Amazon Elastic Beanstalk (Elastic Beanstalk)	Using Elastic Beanstalk API Calls with Amazon CloudTrail	03/31/2014
Amazon Elastic Disaster Recovery	Logging Amazon Elastic Disaster Recovery API calls using Amazon CloudTrail	11/17/2021
AWS Elemental MediaConnect	Logging AWS Elemental MediaConnect API Calls with Amazon CloudTrail	11/27/2018
AWS Elemental MediaConvert	Logging AWS Elemental MediaConvert API Calls with CloudTrail	11/27/2017
AWS Elemental MediaLive	Logging MediaLive API Calls with Amazon CloudTrail	01/19/2019
AWS Elemental MediaPackage	Logging AWS Elemental MediaPackage API Calls with Amazon CloudTrail	12/21/2018
AWS Elemental MediaStore	Logging AWS Elemental MediaStore API Calls with CloudTrail	11/27/2017
AWS Elemental MediaTailor	Logging AWS Elemental MediaTailor API Calls with Amazon CloudTrail	02/11/2019
Amazon Entity Resolution	Logging Amazon Entity Resolution API calls using AAmazon CloudTrail	07/26/2023
Amazon Fault Injection Simulator	Log API calls with Amazon CloudTrail	03/15/2021
Amazon Firewall Manager	Logging Amazon Firewall Manager API Calls with Amazon CloudTrail	04/05/2018
Amazon Global Accelerator	Logging Amazon Global Accelerator API Calls with Amazon CloudTrail	11/26/2018

Amazon Service	CloudTrail Topics	Support began
Amazon Glue	Logging Amazon Glue Operations Using Amazon CloudTrail	11/07/2017
Amazon Ground Station	Logging Amazon Ground Station API Calls with Amazon CloudTrail	05/31/2019
Amazon Health	Logging Amazon Health API Calls with Amazon CloudTrail	11/21/2016
Amazon Health Dashboard	Logging Amazon Health API Calls with Amazon CloudTrail	12/01/2016
Amazon HealthImaging	Logging Amazon HealthImaging API calls using Amazon CloudTrail	07/26/2023
Amazon HealthLake	Logging Amazon HealthLake API calls with Amazon CloudTrail	12/07/2020
Amazon HealthOmics	Logging Amazon HealthOmics API calls using Amazon CloudTrail	11/29/2022
Amazon IAM Identity Center	Logging IAM Identity Center API Calls with Amazon CloudTrail	12/07/2017
Amazon Identity and Access Management (IAM)	Logging IAM Events with Amazon CloudTrail	11/13/2013
Amazon IoT	Logging Amazon IoT API Calls with Amazon CloudTrail	04/11/2016
Amazon IoT 1-Click	Logging Amazon IoT 1-Click API Calls with Amazon CloudTrail	05/14/2018
Amazon IoT Analytics	Logging Amazon IoT Analytics API calls with Amazon CloudTrail	04/23/2018
Amazon IoT Events	Logging Amazon IoT Events API Calls with Amazon CloudTrail	06/11/2019
Amazon IoT Greengrass	Logging Amazon IoT Greengrass API Calls with Amazon CloudTrail	10/29/2018
Amazon IoT Greengrass V2	Log Amazon IoT Greengrass V2 API calls with Amazon CloudTrail	12/14/2020
Amazon IoT SiteWise	Logging Amazon IoT SiteWise API calls with Amazon CloudTrail	04/29/2020
Amazon Key Management Service (Amazon KMS)	Logging Amazon KMS API Calls using Amazon CloudTrail	11/12/2014

Amazon Service	CloudTrail Topics	Support began
Amazon Lake Formation	Logging Amazon Lake Formation API Calls Using Amazon CloudTrail	08/09/2019
Amazon Lambda	Logging Amazon Lambda API Calls By Using Amazon CloudTrail Using Lambda with Amazon CloudTrail	Management events: 04/09/2015 Data events: 11/30/2017
Amazon License Manager	Logging Amazon License Manager API Calls with Amazon CloudTrail	03/01/2019
Amazon Mainframe Modernization	Logging Amazon Mainframe Modernization API calls using Amazon CloudTrail	06/08/2022
Amazon Managed Services	Log management in AMS Accelerate	12/21/2016
Amazon Web Services Marketplace	Logging Amazon Web Services Marketplace API Calls with Amazon CloudTrail	05/02/2017
Amazon Web Services Marketplace Discovery	Logging Amazon Web Services Marketplace Discovery API calls using Amazon CloudTrail	12/15/2022
Amazon Web Services Marketplace Metering Service	Logging Amazon Web Services Marketplace API Calls with Amazon CloudTrail	08/22/2018
Amazon Migration Hub	Logging Amazon Migration Hub API Calls with Amazon CloudTrail	08/14/2017
Amazon Mobile Hub	Logging Amazon Mobile CLI API Calls with Amazon CloudTrail	06/29/2018
Amazon Network Firewall	Logging calls to the Amazon Network Firewall API with Amazon CloudTrail	11/17/2020
Amazon OpsWorks for Chef Automate	Logging Amazon OpsWorks for Chef Automate API Calls with Amazon CloudTrail	07/16/2018
Amazon OpsWorks for Puppet Enterprise	Logging OpsWorks for Puppet Enterprise API Calls with Amazon CloudTrail	07/16/2018
Amazon OpsWorks Stacks	Logging Amazon OpsWorks Stacks API Calls with Amazon CloudTrail	06/04/2014

Amazon Service	CloudTrail Topics	Support began
Amazon Organizations	Logging Amazon Organizations API calls with Amazon CloudTrail	02/27/2017
Amazon Outposts	Logging Amazon Outposts API calls with Amazon CloudTrail	02/04/2020
Amazon Panorama	Amazon Panorama API Reference	10/20/2021
Amazon Payment Cryptography	Logging Amazon Payment Cryptography API calls using Amazon CloudTrail	06/08/2023
Amazon Private 5G	Logging Amazon Private 5G API calls using Amazon CloudTrail	08/11/2022
Amazon Private Certificate Authority	Using CloudTrail	04/04/2018
Amazon Proton	Logging and monitoring in Amazon Proton	06/09/2021
Amazon Resilience Hub	Amazon CloudTrail	11/10/2021
Amazon Resource Access Manager (Amazon RAM)	Logging Amazon RAM API Calls with Amazon CloudTrail	11/20/2018
Amazon Resource Explorer	Logging Amazon Resource Explorer API calls using Amazon CloudTrail	11/07/2022
Amazon Resource Groups	Logging Amazon Resource Groups API Calls with Amazon CloudTrail	06/29/2018
Amazon RoboMaker	Logging Amazon RoboMaker API Calls with Amazon CloudTrail	01/16/2019
Amazon Secrets Manager	Monitor the Use of Your Amazon Secrets Manager Secrets	04/05/2018
Amazon Security Hub	Logging Amazon Security Hub API Calls with Amazon CloudTrail	11/27/2018
Amazon Security Token Service (Amazon STS)	Logging IAM Events with Amazon CloudTrail The IAM topic includes information for Amazon STS.	11/13/2013
Amazon Serverless Application Repository	Logging Amazon Serverless Application Repository API Calls with Amazon CloudTrail	02/20/2018
Amazon Service Catalog	Logging Service Catalog API Calls with Amazon CloudTrail	07/06/2016

Amazon Service	CloudTrail Topics	Support began
Amazon Shield	Logging Shield Advanced API Calls with Amazon CloudTrail	02/08/2018
Amazon Snowball	Logging Amazon Snowball API Calls with Amazon CloudTrail	01/25/2019
Amazon Snowball Edge	Logging Amazon Snowball Edge API Calls with Amazon CloudTrail	01/25/2019
Amazon Step Functions	Logging Amazon Step Functions API Calls with Amazon CloudTrail	12/01/2016
Storage Gateway	Logging Storage Gateway API Calls by Using Amazon CloudTrail	12/16/2014
Amazon Web Services Support	Logging Amazon Web Services Support API Calls with Amazon CloudTrail	04/21/2016
Amazon Systems Manager	Logging Amazon Systems Manager API Calls with Amazon CloudTrail	11/29/2017
Amazon Systems Manager Incident Manager	Logging Amazon Systems Manager Incident Manager API calls using Amazon CloudTrail	05/10/2021
Amazon Telco Network Builder (Amazon TNB)	Logging Amazon Telco Network Builder API calls using Amazon CloudTrail	02/21/2023
Amazon Transfer for SFTP	Logging Amazon Transfer for SFTP API Calls with Amazon CloudTrail	01/08/2019
Amazon Transit Gateway	Logging API Calls for Your Transit Gateway Using Amazon CloudTrail	11/26/2018
Amazon Trusted Advisor	Logging Amazon Trusted Advisor console actions with Amazon CloudTrail	10/22/2020
Amazon Verified Access	Log Amazon Verified Access API calls using Amazon CloudTrail	04/27/2023
Amazon WAF	Logging Amazon WAF API Calls with Amazon CloudTrail	04/28/2016
Amazon Well-Architected Tool	Logging Amazon Well-Architected Tool API Calls with Amazon CloudTrail	12/15/2020
Amazon X-Ray	Logging Amazon X-Ray API Calls With CloudTrail	04/25/2018

Amazon Service	CloudTrail Topics	Support began
Elastic Load Balancing	Amazon CloudTrail Logging for Your Classic Load Balancer and Amazon CloudTrail Logging for Your Application Load Balancer	04/04/2014
FreeRTOS Over-the-Air Updates (OTA)	Logging Amazon IoT OTA API Calls with Amazon CloudTrail	05/22/2019
Service Quotas	Logging Service Quotas API calls using Amazon CloudTrail	06/24/2019

CloudTrail unsupported services

The following Amazon services and events are not supported:

- Amazon Import/Export
- Amazon VPC endpoint policy-specific events

For a list of supported Amazon services, see Amazon service topics for CloudTrail (p. 26).

Quotas in Amazon CloudTrail

The following table describes quotas (formerly referred to as limits) within CloudTrail. CloudTrail has no adjustable quotas. For information about other quotas in Amazon, see Amazon service quotas.

Resource	Default quota	Comments
Trails per Region	5	This quota cannot be increased.
Get, describe, and list APIs	10 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. The `LookupEvents` API is not included in this category. This quota cannot be increased.
LookupEvents API	2 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. This quota cannot be increased.
All other APIs	1 transaction per second (TPS)	The maximum number of operation requests you can make per second without being throttled. This quota cannot be increased.

Resource	Default quota	Comments
Event selectors	5 per trail	This quota cannot be increased.
Advanced event selectors	500 conditions across all advanced event selectors	If a trail uses advanced event selectors, a maximum of 500 total values for all conditions in all advanced event selectors is allowed. Unless a trail logs data events on all resources, such as all S3 buckets or all Lambda functions, you are limited to 250 data resources. Data resources can be distributed across event selectors, but the overall total cannot exceed 250. This quota cannot be increased.

Resource	Default quota	Comments
Data resources in event selectors	250 across all event selectors in a trail	If you choose to limit data events by using event selectors or advanced event selectors, the total number of data resources cannot exceed 250 across all event selectors in a trail. The limit of number of resources on an individual event selector is configurable up to 250. This upper limit is allowed only if the total number of data resources does not exceed 250 across all event selectors. Examples: • A trail with 5 event selectors, each configured with 50 data resources, is allowed. (5*50=250) • A trail with 5 event selectors, 3 of which are configured with 50 data resources, 1 of which is configured with 99 data resources, and 1 of which is configured with 1 data resource, is also allowed. ((3*50)+1+99=250) • A trail configured with 5 event selectors, all of which are configured with 100 data resources, is not allowed. (5*100=500) Event selectors apply only to trails. For event data stores, you must use advanced event selectors. This quota cannot be increased. The quota does not apply if you choose to log data events on all resources, such as all S3 buckets or all Lambda functions.

Resource	Default quota	Comments
Event size	All event versions: events over 256 KB cannot be sent to CloudWatch Logs Event version 1.05 and newer: total event size limit of 256 KB	Amazon CloudWatch Logs and Amazon EventBridge each allow a maximum event size of 256 KB. CloudTrail does not send events over 256 KB to CloudWatch Logs or EventBridge. Starting with event version 1.05, events have a maximum size of 256 KB. This is to help prevent exploitation by malicious actors, and allow events to be consumed by other Amazon services, such as CloudWatch Logs and EventBridge.
CloudTrail file size sent to Amazon S3	50 MB ZIP file, after compression	For both management and data events, CloudTrail sends events to S3 in maximum 50 MB (compressed) ZIP files. If enabled on the trail, log delivery notifications are sent by Amazon SNS after CloudTrail sends ZIP files to S3.

Getting started with Amazon CloudTrail tutorial

If you're new to Amazon CloudTrail, this tutorial helps you learn how to use its features. In this tutorial, you review your recent Amazon account activity in the CloudTrail console and examine an event. You then create a trail, which is an ongoing record of management event activity that is stored in an Amazon S3 bucket. Unlike **Event history**, this ongoing record is not limited to 90 days, logs events in all Amazon Regions, and can help you meet your security and auditing needs over time.

Topics

- Prerequisites (p. 44)
- Step 1: Review Amazon account activity in event history (p. 45)
- Step 2: Create your first trail (p. 45)
- Step 3: View your log files (p. 47)
- Step 4: Plan for next steps (p. 48)

Prerequisites

Before you begin, you must complete the following prerequisites and setup.

Topics

- Sign up for an Amazon Web Services account (p. 44)
- Secure IAM users (p. 44)
- Grant permissions to use the CloudTrail console (p. 45)

Sign up for an Amazon Web Services account

If you do not have an Amazon Web Services account, use the following procedure to create one.

To sign up for Amazon Web Services

1. Open http://www.amazonaws.cn/ and choose **Sign Up**.
2. Follow the on-screen instructions.

Amazon 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 http://www.amazonaws.cn/ and choosing **My Account**.

Secure IAM users

After you sign up for an Amazon Web Services account, safeguard your administrative user by turning on multi-factor authentication (MFA). For instructions, see Enable a virtual MFA device for an IAM user (console) in the *IAM User Guide*.

To give other users access to your Amazon Web Services account resources, create IAM users. To secure your IAM users, turn on MFA and only give the IAM users the permissions needed to perform their tasks.

For more information about creating and securing IAM users, see the following topics in the *IAM User Guide*:

- Creating an IAM user in your Amazon Web Services account
- Access management for Amazon resources
- Example IAM identity-based policies

Grant permissions to use the CloudTrail console

Grant permissions to use the CloudTrail console. For more information, see Granting permissions for CloudTrail administration (p. 355).

Step 1: Review Amazon account activity in event history

CloudTrail is enabled on your Amazon account when you create the account. When activity occurs in any Amazon service that supports CloudTrail, that activity is recorded in a CloudTrail event along with other Amazon service events in **Event history**. In other words, you can view, search, and download recent events in your Amazon account before creating a trail, though creating a trail is important for long-term records and auditing of your Amazon account activity. Unlike a trail, **Event history** only shows events that have occurred over the last 90 days.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Dashboard**. Review the information in your dashboard about the most recent events that have occurred in your Amazon account. A recent event should be a `ConsoleLogin` event, showing that you just signed in to the Amazon Web Services Management Console.
3. To see more information about an event in your dashboard, expand it.
4. In the navigation pane, choose **Event history**. You see a filtered list of events, with the most recent events showing first. The default filter for events is **Read only**, set to **false**. You can clear that filter by choosing **X** at the right of the filter.
5. Many more events are shown without the default filter. You can filter events in many ways. For example, to view all console login events, you could choose the **Event name** filter, and specify **ConsoleLogin**. The choice of filters is up to you.
6. You can save event history by downloading it as a file in CSV or JSON format. Downloading your event history can take a few minutes.

For more information, see Viewing events with CloudTrail Event history (p. 50).

Step 2: Create your first trail

While the events provided in **Event history** in the CloudTrail console are useful for reviewing recent activity, they are limited to recent activity, and they do not include all possible events that can be recorded by CloudTrail. Additionally, your view of events in the console is limited to the Amazon Region where you are signed in. To create an ongoing record of activity in your Amazon account that captures information for all Amazon Regions, create a trail. By default, when you create a trail in the CloudTrail console, the trail logs events in all Regions. Logging events in all Regions in your account is a recommended best practice.

For your first trail, we recommend creating a trail that logs all management events (p. 6) in all Amazon Regions, and does not log any data events (p. 6). Examples of management events include security events such as IAM `CreateUser` and `AttachRolePolicy` events, resource events such as `RunInstances` and `CreateBucket`, and many more. You will create an Amazon S3 bucket where you will store the log files for the trail as part of creating the trail in the CloudTrail console.

Note
This tutorial assumes you are creating your first trail. Depending on the number of trails you have in your Amazon account, and how those trails are configured, the following procedure might or might not incur expenses. CloudTrail stores log files in an Amazon S3 bucket, which incurs costs. For more information about pricing, see Amazon CloudTrail Pricing and Amazon S3 Pricing.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the **Region** selector, choose the Amazon Region where you want your trail to be created. This is the home Region for the trail.

   Note
   The home Region is the only Amazon Region where you can view and update the trail after it is created, even if the trail logs events in all Amazon Regions.
3. On the CloudTrail service home page, the **Trails** page, or the **Trails** section of the **Dashboard** page, choose **Create trail**.
4. In **Trail name**, give your trail a name, such as *`My-Management-Events-Trail`*. As a best practice, use a name that quickly identifies the purpose of the trail. In this case, you're creating a trail that logs management events.
5. Leave default settings for Amazon Organizations organization trails. This option won't be available to change unless you have accounts configured in Organizations.
6. For **Storage location**, choose **Create new S3 bucket** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies. Give your bucket a name, such as *`my-bucket-for-storing-cloudtrail-logs`*.

   To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.

   Note
   The name of your Amazon S3 bucket must be globally unique. For more information, see Bucket naming rules in the *Amazon Simple Storage Service User Guide*.
7. Clear the check box to disable **Log file SSE-KMS encryption**. By default, your log files are encrypted with SSE-S3 encryption. For more information about this setting, see Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).
8. Leave default settings in **Additional settings**.
9. For now, do not send logs to Amazon CloudWatch Logs.
10. In **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify your CloudTrail trails and other resources, such as the Amazon S3 buckets that contain CloudTrail log files. For example, you could attach a tag with the name **`Compliance`** and the value **`Auditing`**.

    Note
    Though you can add tags to trails when you create them in the CloudTrail console, and you can create an Amazon S3 bucket to store your log files in the CloudTrail console, you cannot add tags to the Amazon S3 bucket from the CloudTrail console. For more information about viewing and changing the properties of an Amazon S3 bucket, including adding tags to a bucket, see the *Amazon S3 User Guide*.

    When you are finished creating tags, choose **Next**.
11. On the **Choose log events** page, select event types to log. For this trail, keep the default, **Management events**. In the **Management events** area, choose to log both **Read** and **Write** events,

if they are not already selected. Leave the check box for **Exclude Amazon KMS events** empty, to log all events.

12. Leave default settings for **Data events** and Insights events. This trail will not log any data or CloudTrail Insights events. Choose **Next**.
13. On the **Review and create** page, review the settings you've chosen for your trail. Choose **Edit** for a section to go back and make changes. When you are ready to create your trail, choose **Create trail**.
14. The **Trails** page shows your new trail in the table. Note that the trail is set to **Multi-region trail** by default, and that logging is turned on for the trail by default.

Step 3: View your log files

Within an average of about 5 minutes of creating your first trail, CloudTrail delivers the first set of log files to the Amazon S3 bucket for your trail. You can look at these files and learn about the information they contain.

Note
CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed. Review the Amazon CloudTrail Service Level Agreement for more information.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Trails**. On the **Trails** page, find the name of the trail you just created (in the example, *My-Management-Events-Trail*).
3. In the row for the trail, choose the value for the S3 bucket (in the example, *aws-cloudtrail-logs-08132020-mytrail*).
4. The Amazon S3 console opens and shows that bucket, at the top level for log files. Because you created a trail that logs events in all Amazon Regions, the display opens at the level that shows you each Region folder. The hierarchy of the Amazon S3 bucket navigation at this level is *bucket-name*/AmazonLogs/*account-id*/CloudTrail. Choose the folder for the Amazon Region where you want to review log files. For example, if you want to review the log files for the US East (Ohio) Region, choose **us-east-2**.
5. Navigate the bucket folder structure to the year, the month, and the day where you want to review logs of activity in that Region. In that day, there are a number of files. The name of the files begin with your Amazon account ID, and end with the extension `.gz`. For example, if your account ID is *123456789012*, you would see files with names similar to this: *123456789012*_CloudTrail_*us-east-2*_*20190610T1255abcdeEXAMPLE*.json.gz.

   To view these files, you can download them, unzip them, and then view them in a plain-text editor or a JSON file viewer. Some browsers also support viewing .gz and JSON files directly. We recommend using a JSON viewer, as it makes it easier to parse the information in CloudTrail log files.

   As you're browsing through the file content, you might start to wonder about what you're seeing. CloudTrail logs events for every Amazon service that experienced activity in that Amazon Region at the time that event occurred. In other words, events for different Amazon services are mixed together, based solely on time. To learn more about what a specific Amazon service logs with CloudTrail, including examples of log file entries for API calls for that service, see the list of supported services for CloudTrail (p. 26), and read the CloudTrail integration topic for that service. You can also learn more about the content and structure of CloudTrail log files by reviewing the CloudTrail log event reference (p. 400).

   You might also notice what you're not seeing in log files in US East (Ohio). Specifically, you won't see any console sign-in events, even though you know you logged into the console. That's because console sign-in and IAM events are global service events (p. 14), which are usually logged in a specific Amazon Region. In this case, they are logged in US East (N. Virginia), and found in the folder

us-east-1. Open that folder, and open the year, month, and day you're interested in. Browse the log files, and you find `ConsoleLogin` events that look similar to the following:


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Mary_Major",
        "accountId": "123456789012",
        "userName": "Mary_Major"
    },
    "eventTime": "2019-06-10T17:14:09Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.67",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101
 Firefox/60.0",
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Success"
    },
    "additionalEventData": {
        "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
        "MobileVersion": "No",
        "MFAUsed": "No"
    },
    "eventID": "2681fc29-EXAMPLE",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "123456789012"
}
```


This log file entry tells you more than just the identity of the IAM user who logged in (`Mary_Major`), the date and time she logged in, and that the login was successful. You can also learn the IP address she logged in from, the operating system and browser software of the computer she used, and that she was not using multi-factor authentication.

Step 4: Plan for next steps

Now that you have a trail, you have access to an ongoing record of events and activities in your Amazon account. This ongoing record helps you meet accounting and auditing needs for your Amazon account. However, there is a lot more you can do with CloudTrail and CloudTrail data.

- **Add additional security for your trail data.** CloudTrail automatically applies a certain level of security when you create a trail. However, there are additional steps you can take to help keep your data secure.
  - By default, the Amazon S3 bucket you created as part of creating a trail has a policy applied that allows CloudTrail to write log files to that bucket. The bucket is not publicly accessible, but it might be accessible to other users in your Amazon account if they have permissions to read and write to buckets in your Amazon account. Review the policy for your bucket and if necessary, make changes to restrict access. For more information, see the Amazon S3 security documentation and the example walkthrough for securing a bucket.
  - The log files delivered by CloudTrail to your bucket are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). To provide a security layer that is directly manageable, you can instead use server-side encryption with Amazon KMS–managed keys (SSE-KMS) for your CloudTrail log files. To use SSE-KMS with CloudTrail, you create and manage a KMS

key, also known as an Amazon KMS key. For more information, see Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386).

- For additional security planning, review the security best practices for CloudTrail (p. 382).

- **Create a trail to log data events.** If you are interested in logging when objects are added, retrieved, and deleted in one or more Amazon S3 buckets, when items are added, changed, or deleted in DynamoDB tables, or when one or more Amazon Lambda functions are invoked, these are data events. The management event trail you created earlier in this tutorial doesn't log these types of events. You can create a separate trail specifically to log data events for some or all of supported resources. For more information, see Data events (p. 243).

  Note
  Additional charges apply for logging data events. For more information, see Amazon CloudTrail Pricing.

- **Log CloudTrail Insights events on your trail.** CloudTrail Insights helps you identify and respond to unusual or anomalous activity associated with `write` API calls by continuously analyzing CloudTrail management events. CloudTrail Insights uses mathematical models to determine the normal levels of API and service event activity for an account. It identifies behavior that is outside normal patterns, generates Insights events, and delivers those events to a `/CloudTrail-Insight` folder in the chosen destination S3 bucket for your trail. For more information about CloudTrail Insights, see Logging Insights events for trails (p. 267).

  Note
  Additional charges apply for logging Insights events. For more information, see Amazon CloudTrail Pricing.

- **Set up CloudWatch Logs alarms to alert you when certain events occur.** CloudWatch Logs lets you monitor and receive alerts for specific events captured by CloudTrail. For example, you can monitor key security and network-related management events, such as security group changes (p. 283), failed Amazon Web Services Management Console sign-in events (p. 284), or changes to IAM policies (p. 285). For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).
- **Use analysis tools to identify trends in your CloudTrail logs.** While the filters in Event history can help you find specific events or event types in your recent activity, it does not provide the ability to search through activity over longer time periods. For deeper and more sophisticated analysis, you can use Amazon Athena. For more information, see Querying Amazon CloudTrail Logs in the Amazon Athena User Guide.

Working with CloudTrail

CloudTrail is enabled by default for your Amazon account. You can use **Event history** in the CloudTrail console to view, search, download, archive, analyze, and respond to account activity across your Amazon infrastructure. This includes activity made through the Amazon Web Services Management Console, Amazon Command Line Interface, and Amazon SDKs and APIs.

For an ongoing record of events in your Amazon account, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Amazon Regions. The trail logs events from all Regions in the Amazon partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other Amazon services to further analyze and act upon the event data collected in CloudTrail logs.

If you have created an organization in Amazon Organizations, you can create a trail that will log all events for all Amazon accounts in that organization. Creating an organization trail helps you define a uniform event logging strategy for your organization.

Topics

- Viewing events with CloudTrail Event history (p. 50)
- Viewing CloudTrail Insights events (p. 62)
- Creating a trail for your Amazon Web Services account (p. 77)
- Creating a trail for an organization (p. 125)
- Copying trail events to CloudTrail Lake (p. 139)
- Getting and viewing your CloudTrail log files (p. 144)
- Configuring Amazon SNS notifications for CloudTrail (p. 147)
- Controlling user permissions for CloudTrail (p. 148)
- Tips for managing trails (p. 149)
- Using Amazon CloudTrail with interface VPC endpoints (p. 152)
- Viewing service-linked channels for CloudTrail by using the Amazon CLI (p. 153)

Viewing events with CloudTrail Event history

You can troubleshoot operational and security incidents in the CloudTrail console by viewing **Event history**. The **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded API activity (management events) in an Amazon Web Services Region.

You can look up events related to the creation, modification, or deletion of resources (such as IAM users or Amazon EC2 instances) in your Amazon Web Services account on a by-Region basis. You can view events and download them with the Amazon CloudTrail console. You can customize the view of event history in the console by selecting which columns to display or hide. You can programmatically look up events by using the Amazon SDKs or Amazon Command Line Interface. You can also compare the details of events in **Event history** side-by-side.

CloudTrail logging varies between Amazon services. While most Amazon services support CloudTrail logging of all events, some services only support logging a subset of APIs and events, and a few services

are unsupported. For example, some Amazon services log a `ThrottlingException` when API requests are throttled, while other Amazon services do not. You can learn more about the specifics of how CloudTrail logs events for a specific service by consulting the documentation for that service. For more information, see CloudTrail supported services and integrations (p. 24).

> **Note**
> Over time, Amazon Web Services might add additional events. CloudTrail records these events in **Event history**, but a full 90-day record of activity that includes added events won't be available until 90 days after it adds the events.

This section describes how to look up events by using the CloudTrail console and the Amazon CLI. It also describes how to download a file of events. For information on using the `LookupEvents` API to retrieve information from CloudTrail events, see the Amazon CloudTrail API Reference.

For information on creating a trail so that you have a record of events that extends past 90 days, see Creating a trail (p. 78) and Getting and viewing your CloudTrail log files (p. 144).

Topics

- Viewing CloudTrail events in the CloudTrail console (p. 51)
- Viewing CloudTrail events with the Amazon CLI (p. 55)

Viewing CloudTrail events in the CloudTrail console

You can use the CloudTrail console to view the last 90 days of recorded API activity (management events) in an Amazon Region. You can also download a file with that information, or a subset of information based on the filter and time range you choose. You can customize your view of **Event history** by selecting which columns are displayed in the console. You can also look up and filter events by the resource types available for a particular service. You can select up to five events in **Event history** and compare their details side-by-side.

Event history does not show data events. To view data events, create a trail (p. 77).

After 90 days, events are no longer shown in **Event history**. You cannot manually delete events from **Event history**. When you create a trail (p. 78), you can view events that are logged to your trail for as long as you store them in the S3 bucket that is configured in your trail settings.

CloudTrail logging varies between Amazon services. While most Amazon services support CloudTrail logging of all events, some services only support logging a subset of APIs and events, and a few services are unsupported. You can learn more about the specifics of how CloudTrail logs events for a specific service by consulting the documentation for that service. For more information, see CloudTrail supported services and integrations (p. 24).

> **Note**
> For an ongoing record of activity and events, create a trail (p. 78). Creating a trail also enables you to take advantage of the following integrations:
>
> - A trail lets you log CloudTrail Insights events, which can help you identify and respond to unusual activity associated with `write` management API calls. For more information, see Logging Insights events for trails (p. 267).
> - Analyze your Amazon service activity with queries in Amazon Athena. For more information, see Creating a Table for CloudTrail Logs in the CloudTrail Console in the Amazon Athena User Guide, or choose the option to create a table directly from **Event history** in the CloudTrail console.
> - Monitor your trail logs and be notified when specific activity occurs with Amazon CloudWatch Logs. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

- A trail lets you exclude Amazon Key Management Service (Amazon KMS) or Amazon Relational Database Service Data API events. Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. Events cannot be excluded from **Event history**; you can only exclude events if you create or update a trail to log management events.

To view CloudTrail events

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/home/.
2. In the navigation pane, choose **Event history**.

   A filtered list of events appears in the content pane with the latest event first. Scroll down to see more events.
3. To compare events, select up to five events by filling their check boxes in the left margin of the **Event history** table. View details for selected events side-by-side in the **Compare event details** table.

The default view of events in **Event history** has a filter applied so that it does not display read-only events. To remove this filter, or to apply other filters, change the filter settings. For more information, see Filtering CloudTrail events (p. 53).

Contents

- Displaying CloudTrail events (p. 52)
- Filtering CloudTrail events (p. 53)
- Viewing details for an event (p. 54)
- Downloading events (p. 54)
- Viewing resources referenced with Amazon Config (p. 55)

Displaying CloudTrail events

You can customize the display of **Event history** by selecting which columns to display in the CloudTrail console. By default, the following columns are displayed:

- **Event name**
- **Event time**
- **User name**
- **Event source**
- **Resource type**
- **Resource name**

> **Note**
> You cannot change the order of the columns, or manually delete events from **Event history**.

To customize the columns displayed in Event history

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/home/.
2. In the navigation pane, choose **Event history**.
3. Choose the gear icon.

4. In **Select visible columns**, select the columns you want to display. Turn off columns you do not want to display. When you have finished, choose **Confirm**.

Filtering CloudTrail events

The default display of events in **Event history** uses an attribute filter to exclude read-only events from the list of displayed events. This attribute filter is named **Read-only**, and it is set to **false**. You can remove this filter to display both read and write events. To view only **Read** events, you can change the filter value to **true**. You can also filter events by other attributes. You can additionally filter by time range.

> **Note**
> You can only apply one attribute filter and a time range filter. You cannot apply multiple attribute filters.

Amazon access key

The Amazon access key ID that was used to sign the request. If the request was made with temporary security credentials, this is the access key ID of the temporary credentials.

Event ID

The CloudTrail ID of the event. Each event has a unique ID.

Event name

The name of the event. For example, you can filter on IAM events, such as `CreatePolicy`, or Amazon EC2 events, such as `RunInstances`.

Event source

The Amazon service to which the request was made, such as `iam.amazonaws.com` or `s3.amazonaws.com`. You can scroll through a list of event sources after you choose the **Event source** filter.

Read only

The read type of the event. Events are categorized as read events or write events. If set to **false**, read events are not included in the list of displayed events. By default, this attribute filter is applied and the value is set to **false**.

Resource name

The name or ID of the resource referenced by the event. For example, the resource name might be "auto-scaling-test-group" for an Auto Scaling group or "i-12345678910" for an EC2 instance.

Resource type

The type of resource referenced by the event. For example, a resource type can be `Instance` for EC2 or `DBInstance` for RDS. Resource types vary for each Amazon service.

Time range

The time range in which you want to filter events. You can filter events for the last 90 days.

User name

The identity referenced by the event. For example, this can be a user, a role name, or a service role.

If there are no events logged for the attribute or time that you choose, the results list is empty. You can apply only one attribute filter in addition to the time range. If you choose a different attribute filter, your specified time range is preserved.

The following steps describe how to filter by attribute.

To filter by attribute

1. To filter the results by an attribute, choose an attribute from the **Lookup attributes** drop-down list, and then type or choose a value for the attribute in the text box.
2. To remove an attribute filter, choose the **X** at the right of the attribute filter box.

The following steps describe how to filter by a start and end date and time.

To filter by a start and end date and time

1. To narrow the time range for the events that you want to see, choose a time range in the time range bar. Preset values are 30 minutes, 1 hour, 3 hours, or 12 hours. To specify a custom time range, choose **Custom**.
2. To remove a time range filter, choose **Clear** in the time range bar.

Viewing details for an event

1. Choose an event in the results list to show its details.
2. Resources referenced in the event are shown in the **Resources referenced** table on the event details page.
3. Some referenced resources have links. Choose the link to open the console for that resource.
4. Scroll to **Event record** on the details page to see the JSON event record, also called the event *payload*.
5. Choose **Event history** in the page breadcrumb to close the event details page and return to **Event history**.

Downloading events

You can download recorded event history as a file in CSV or JSON format. Use filters and time ranges to reduce the size of the file you download.

> **Note**
> CloudTrail event history files are data files that contain information (such as resource names) that can be configured by individual users. Some data can potentially be interpreted as commands in programs used to read and analyze this data (CSV injection). For example, when CloudTrail events are exported to CSV and imported to a spreadsheet program, that program might warn you about security concerns. You should choose to disable this content to keep your system secure. Always disable links or macros from downloaded event history files.

1. Add a filter and time range for events in **Event history** that you want to download. For example, you can specify the event name, `StartInstances`, and specify a time range for the last three days of activity.
2. Choose **Download events**, and then choose **Download as CSV** or **Download as JSON**. The download starts immediately.

   > **Note**
   > Your download might take some time to complete. For faster results, before you start the download process, use a more specific filter or a shorter time range to narrow the results. You can cancel a download. If you cancel a download, a partial download including only some event data might be on your local computer. To download the full event history, restart the download.
3. After your download is complete, open the file to view the events that you specified.
4. To cancel your download, choose **Cancel**, and then confirm by choosing **Cancel download**. If you need to restart a download, wait until the earlier download is finished canceling.

Viewing resources referenced with Amazon Config

Amazon Config records configuration details, relationships, and changes to your Amazon resources.

On the **Resources Referenced** pane, choose the  in the **Config timeline** column to view the resource in the Amazon Config console.

If the  icon is gray, Amazon Config isn't turned on, or it's not recording the resource type. Choose the icon to go to the Amazon Config console to turn on the service or start recording that resource type. For more information, see Set Up Amazon Config Using the Console in the *Amazon Config Developer Guide*.

If **Link not available** appears in the column, the resource can't be viewed for one of the following reasons:

- Amazon Config doesn't support the resource type. For more information, see Supported Resources, Configuration Items, and Relationships in the *Amazon Config Developer Guide*.
- Amazon Config recently added support for the resource type, but it's not yet available from the CloudTrail console. You can look up the resource in the Amazon Config console to see the timeline for the resource.
- The resource is owned by another Amazon account.
- The resource is owned by another Amazon service, such as a managed IAM policy.
- The resource was created and then deleted immediately.
- The resource was recently created or updated.

Example

1. You configure Amazon Config to record IAM resources.
2. You create an IAM user, **`Bob-user`**. The **Event history** page shows the `CreateUser` event and **`Bob-user`** as an IAM resource. You can choose the Amazon Config icon to view this IAM resource in the Amazon Config timeline.
3. You update the user name to **`Bob-admin`**.
4. The **Event history** page shows the `UpdateUser` event and **`Bob-admin`** as the updated IAM resource.
5. You can choose the icon to view the **`Bob-admin`** IAM resource in the timeline. However, you can't choose the icon for **`Bob-user`**, because the resource name changed. Amazon Config is now recording the updated resource.

To grant users read-only permission to view resources in the Amazon Config console, see Granting permission to view Amazon Config information on the CloudTrail console (p. 360).

For more information about Amazon Config, see the Amazon Config Developer Guide.

Viewing CloudTrail events with the Amazon CLI

You can look up CloudTrail management events for the last 90 days for the current Amazon Web Services Region using the **aws cloudtrail lookup-events** command. The **aws cloudtrail lookup-events** command shows events in the Amazon Web Services Region where they occurred.

Lookup supports the following attributes for management events:

- Amazon access key
- Event ID
- Event name
- Event source

- Read only
- Resource name
- Resource type
- User name

All attributes are optional.

The `lookup-events` command includes the following options:

- `--max-items` *<integer>* – The total number of items to return in the command's output. If the total number of items available is more than the value specified, a `NextToken` is provided in the command's output. To resume pagination, provide the `NextToken` value in the starting-token argument of a subsequent command. Do not use the `NextToken` response element directly outside of the Amazon CLI.
- `--start-time` *<timestamp>* – Specifies that only events that occur after or at the specified time are returned. If the specified start time is after the specified end time, an error is returned.
- `--lookup-attributes` *<integer>* – Contains a list of lookup attributes. Currently the list can contain only one item.
- `--generate-cli-skeleton` *<string>* – Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for `--cli-input-json`. Similarly, if provided yaml-input it will print a sample input YAML that can be used with `--cli-input-yaml`. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command. The generated JSON skeleton is not stable between versions of the Amazon CLI and there are no backwards compatibility guarantees in the JSON skeleton generated.
- `--cli-input-json` *<string>* – Reads arguments from the JSON string provided. The JSON string follows the format provided by the `--generate-cli-skeleton` parameter. If other arguments are provided on the command line, those values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally. This may not be specified along with the `--cli-input-yaml` parameter.

For general information about using the Amazon Command Line Interface, see the Amazon Command Line Interface User Guide.

Contents

- Prerequisites (p. 56)
- Getting command line help (p. 57)
- Looking up events (p. 57)
- Specifying the number of events to return (p. 58)
- Looking up events by time range (p. 58)
    - Valid <timestamp> formats (p. 58)
- Looking up events by attribute (p. 58)
    - Attribute lookup examples (p. 59)
- Specifying the next page of results (p. 60)
- Getting JSON input from a file (p. 60)
- Lookup output fields (p. 61)

Prerequisites

- To run Amazon CLI commands, you must install the Amazon CLI. For information, see Installing the Amazon Command Line Interface.

- Make sure your Amazon CLI version is greater than 1.6.6. To verify the CLI version, run **aws --version** on the command line.
- To set the account, Amazon Web Services Region, and default output format for an Amazon CLI session, use the **aws configure** command. For more information, see Configuring the Amazon Command Line Interface.

Note
The CloudTrail Amazon CLI commands are case-sensitive.

Getting command line help

To see the command line help for `lookup-events`, type the following command:

```
aws cloudtrail lookup-events help
```

Looking up events

Important
The rate of lookup requests is limited to two per second, per account, per Region. If this limit is exceeded, a throttling error occurs.

To see the ten latest events, type the following command:

```
aws cloudtrail lookup-events --max-items 10
```

A returned event looks similar to the following fictitious example, which has been formatted for readability:

```
{
    "NextToken": "kbOt5LlZe+
+mErCebpy2TgaMgmDvF1kYGFcH64JSjIbZFjsuvrSqg66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXd12juy3CIZW8=",
    "Events": [
        {
            "EventId": "0ebbaee4-6e67-431d-8225-ba0d81df5972",
            "Username": "root",
            "EventTime": 1424476529.0,
            "CloudTrailEvent": "{
                  \"eventVersion\":\"1.02\",
                  \"userIdentity\":{
                        \"type\":\"Root\",
                        \"principalId\":\"111122223333\",
                        \"arn\":\"arn:aws:iam::111122223333:root\",
                        \"accountId\":\"111122223333\"},
                  \"eventTime\":\"2015-02-20T23:55:29Z\",
                  \"eventSource\":\"signin.amazonaws.com\",
                  \"eventName\":\"ConsoleLogin\",
                  \"awsRegion\":\"us-east-2\",
                  \"sourceIPAddress\":\"203.0.113.4\",
                  \"userAgent\":\"Mozilla/5.0\",
                  \"requestParameters\":null,
                  \"responseElements\":{\"ConsoleLogin\":\"Success\"},
                  \"additionalEventData\":{
                         \"MobileVersion\":\"No\",
                         \"LoginTo\":\"https://console.aws.amazon.com/console/home",
                         \"MFAUsed\":\"No\"},
                  \"eventID\":\"0ebbaee4-6e67-431d-8225-ba0d81df5972\",
                  \"eventType\":\"AwsApiCall\",
                  \"recipientAccountId\":\"111122223333\"}",
            "EventName": "ConsoleLogin",
```

```
            "Resources": []
        }
    ]
}
```

For an explanation of the lookup-related fields in the output, see the section Lookup output fields (p. 61) later in this document. For an explanation of the fields in the CloudTrail event, see CloudTrail record contents (p. 402).

Specifying the number of events to return

To specify the number of events to return, type the following command:

```
aws cloudtrail lookup-events --max-items <integer>
```

Possible values are 1 through 50. The following example returns one event.

```
aws cloudtrail lookup-events --max-items 1
```

Looking up events by time range

Events from the past 90 days are available for lookup. To specify a time range, type the following command:

```
aws cloudtrail lookup-events --start-time <timestamp> --end-time <timestamp>
```

`--start-time` *<timestamp>* specifies that only events that occur after or at the specified time are returned. If the specified start time is after the specified end time, an error is returned.

`--end-time` *<timestamp>* specifies that only events that occur before or at the specified time are returned. If the specified end time is before the specified start time, an error is returned.

The default start time is the earliest date that data is available within the last 90 days. The default end time is the time of the event that occurred closest to the current time.

Valid *<timestamp>* formats

The `--start-time` and `--end-time` attributes take UNIX time values or valid equivalents.

The following are examples of valid formats. Date, month, and year values can be separated by hyphens or forward slashes. Double quotes must be used if spaces are present.

```
1422317782
1422317782.0
01-27-2015
01-27-2015,01:16PM
"01-27-2015, 01:16 PM"
"01/27/2015, 13:16"
2015-01-27
"2015-01-27, 01:16 PM"
```

Looking up events by attribute

To filter by an attribute, type the following command:

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=<attribute>,AttributeValue=<string>
```

You can specify only one attribute key/value pair for each **lookup-events** command. The following are values for `AttributeKey`. Value names are case sensitive.

- AccessKeyId
- EventId
- EventName
- EventSource
- ReadOnly
- ResourceName
- ResourceType
- Username

Attribute lookup examples

The following example command returns events in Region `cn-northwest-1`, which allows you to view global service events. Replace *`gseService`* with the appropriate value for your configuration.

```
aws cloudtrail --region cn-northwest-1  lookup-events --lookup-attributes
 AttributeKey=EventName,AttributeValue=gseService
```

The following example command returns events in which the value of `AccessKeyId` is `AKIAIOSFODNN7EXAMPLE`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=AccessKeyId,AttributeValue=AKIAIOSFODNN7EXAMPLE
```

The following example command returns the event for the specified CloudTrail `EventId`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=EventId,AttributeValue=b5cc8c40-12ba-4d08-a8d9-2bceb9a3e002
```

The following example command returns events in which the value of `EventName` is `RunInstances`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=EventName,AttributeValue=RunInstances
```

The following example command returns events in which the value of `EventSource` is `iam.amazonaws.com`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=EventSource,AttributeValue=iam.amazonaws.com
```

The following example command returns write events. It excludes read events such as `GetBucketLocation` and `DescribeStream`.

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=ReadOnly,AttributeValue=false
```

The following example command returns events in which the value of `ResourceName` is `CloudTrail_CloudWatchLogs_Role`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceName,AttributeValue=CloudTrail_CloudWatchLogs_Role
```

The following example command returns events in which the value of `ResourceType` is `AWS::S3::Bucket`.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceType,AttributeValue=AWS::S3::Bucket
```

The following example command returns events in which the value of `Username` is `root`.

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=Username,AttributeValue=root
```

Specifying the next page of results

To get the next page of results from a `lookup-events` command, type the following command:

```
aws cloudtrail lookup-events <same parameters as previous command> --next-token=<token>
```

where the value for *<token>* is taken from the first field of the output of the previous command.

When you use `--next-token` in a command, you must use the same parameters as in the previous command. For example, suppose you run the following command:

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=Username,AttributeValue=root
```

To get the next page of results, your next command would look like this:

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=Username,AttributeValue=root
 --next-token=kbOt5LlZe+
+mErCebpy2TgaMgmDvF1kYGFcH64JSjIbZFjsuvrSqg66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXd12juy3CIZW8=
```

Getting JSON input from a file

The Amazon CLI for some Amazon services has two parameters, `--generate-cli-skeleton` and `--cli-input-json`, that you can use to generate a JSON template which you can modify and use as input to the `--cli-input-json` parameter. This section describes how to use these parameters with `aws cloudtrail lookup-events`. For more general information, see Generate CLI Skeleton and CLI Input JSON Parameters.

To look up CloudTrail events by getting JSON input from a file

1. Create an input template for use with `lookup-events` by redirecting the `--generate-cli-skeleton` output to a file, as in the following example.

   ```
   aws cloudtrail lookup-events --generate-cli-skeleton > LookupEvents.txt
   ```

   The template file generated (in this case, LookupEvents.txt) looks like this:

   ```
   {
       "LookupAttributes": [
           {
               "AttributeKey": "",
               "AttributeValue": ""
           }
       ],
       "StartTime": null,
   ```

```
    "EndTime": null,
    "MaxResults": 0,
    "NextToken": ""
}
```

2. Use a text editor to modify the JSON as needed. The JSON input must contain only values that are specified.

   Important
   All empty or null values must be removed from the template before you can use it.

   The following example specifies a time range and maximum number of results to return.

```
{
    "StartTime": "2015-01-01",
    "EndTime": "2015-01-27",
    "MaxResults": 2
}
```

3. To use the edited file as input, use the syntax `--cli-input-json file://<filename>`, as in the following example:

```
aws cloudtrail lookup-events --cli-input-json file://LookupEvents.txt
```

   Note
   You can use other arguments on the same command line as `--cli-input-json` .

Lookup output fields

Events

A list of lookup events based on the lookup attribute and time range that were specified. The events list is sorted by time, with the latest event listed first. Each entry contains information about the lookup request and includes a string representation of the CloudTrail event that was retrieved.

The following entries describe the fields in each lookup event.

CloudTrailEvent

A JSON string that contains an object representation of the event returned. For information about each of the elements returned, see Record Body Contents.

EventId

A string that contains the GUID of the event returned.

EventName

A string that contains the name of the event returned.

EventSource

The Amazon service that the request was made to.

EventTime

The date and time, in UNIX time format, of the event.

Resources

A list of resources referenced by the event that was returned. Each resource entry specifies a resource type and a resource name.

ResourceName

A string that contains the name of the resource referenced by the event.

ResourceType

A string that contains the type of a resource referenced by the event. When the resource type cannot be determined, null is returned.

Username

A string that contains the user name of the account for the event returned.

NextToken

A string to get the next page of results from a previous `lookup-events` command. To use the token, the parameters must be the same as those in the original command. If no `NextToken` entry appears in the output, there are no more results to return.

Viewing CloudTrail Insights events

After you enable CloudTrail Insights on a trail, you can view up to 90 days of Insights events by using the CloudTrail console or the Amazon CLI. This section describes how to view, look up, and download a file of Insights events. For information about using the `LookupEvents` API to retrieve information from CloudTrail events, see the Amazon CloudTrail API Reference. For more information about CloudTrail Insights, see Logging Insights events for trails (p. 267) in this guide.

For information about how to create a trail, see Creating a trail (p. 78) and Getting and viewing your CloudTrail log files (p. 144).

> **Note**
> To log Insights events on API call volume, the trail must log `write` management events. To log Insights events on API error rate, the trail must log `read` or `write` management events.

Topics

- Viewing CloudTrail Insights events in the CloudTrail console (p. 62)
- Viewing CloudTrail Insights events with the Amazon CLI (p. 69)

Viewing CloudTrail Insights events in the CloudTrail console

After you enable CloudTrail Insights events on a trail, when CloudTrail detects unusual API or error rate activity, CloudTrail generates Insights events and displays them on the **Dashboard** and **Insights** pages in the Amazon Web Services Management Console. You can view the Insights events in the console and troubleshoot the unusual activity. The most recent 90 days of Insights events are shown in the console. You can also download Insights events by using the Amazon CloudTrail console. You can programmatically look up events by using the Amazon SDKs or Amazon Command Line Interface. For more information about CloudTrail Insights events, see Logging Insights events for trails (p. 267) in this guide.

> **Note**
> To log Insights events on API call volume, the trail must log `write` management events. To log Insights events on API error rate, the trail must log `read` or `write` management events.

After Insights events are logged, the events are shown on the **Insights** page for 90 days. You cannot manually delete events from the **Insights** page. Because you must create a trail (p. 78) before you can

enable CloudTrail Insights, you can view Insights events that are logged to your trail for as long as you store them in the S3 bucket that is configured in your trail settings.

Monitor your trail logs and be notified when specific Insights events activity occurs with Amazon CloudWatch Logs. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

To view Insights events

CloudTrail Insights events must be enabled on your trail to see Insights events in the console. Allow up to 36 hours for CloudTrail to deliver the first Insights events, if unusual activity is detected.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/home/.
2. In the navigation pane, choose **Dashboard** to see the five most recent Insights events, or **Insights** to see all Insights events logged in your account in the last 90 days.

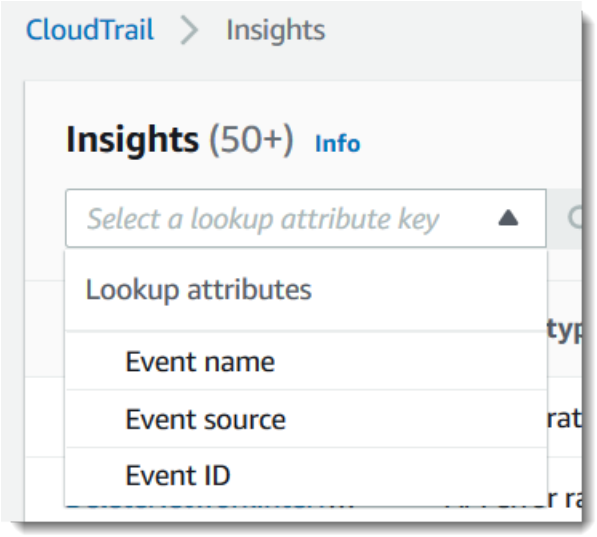
   On the **Insights** page, you can filter Insights events by criteria including event API source, event name, and event ID, and limit the events displayed to those occurring within a specific time range. For more information about filtering Insights events, see Filtering Insights events (p. 63).

Contents

- Filtering Insights events (p. 63)
- Viewing Insights events details (p. 54)
- Zoom, pan, and download graph (p. 67)
- Change graph time span settings (p. 67)
- Downloading Insights events (p. 69)

Filtering Insights events

The default display of events in **Insights** shows events in reverse chronological order. The newest Insights events, sorted by event start time, are at the top. The following list describes the available attributes. You can filter on the first three attributes: **Event name**, **Event source**, and **Event ID**.



Event name

The name of the event, typically the Amazon API on which unusual levels of activity were recorded.

Insight type

The type of CloudTrail Insights event, which is either **API call rate** or **API error rate**. The **API call rate** insight type analyzes write-only management API calls that are aggregated per minute against a baseline API call volume. The **API error rate** insight type analyzes management API calls that result in error codes. The error is shown if the API call is unsuccessful.

Event source

The Amazon service to which the request was made, such as `iam.amazonaws.com` or `s3.amazonaws.com`. You can scroll through a list of event sources after you choose the **Event source** filter.

Event ID

The ID of the Insights event. Event IDs are not shown in the **Insights** page table, but they are an attribute on which you can filter Insights events. The event IDs of management events that are analyzed to generate Insights events are different from the event IDs of Insights events.

Event start time

The start time of the Insights event, measured as the first minute in which unusual activity was recorded. This attribute is shown in the **Insights** table, but you cannot filter on event start time in the console.

Baseline average

The normal pattern of API call rate or error rate activity. The baseline average is calculated over the seven days preceding the start of an Insights event. Though the value of the baseline duration—the period that CloudTrail analyzes for normal activity on APIs—is approximately seven days, CloudTrail rounds the baseline duration to a whole integer day, so the exact baseline duration can vary.

Insight average

The average number of calls to an API, or the average number of a specific error that was returned on calls to an API, that triggered the Insights event. The CloudTrail Insights average for the start event is the rate of occurrences that triggered the Insights event. Typically, this is the first minute of unusual activity. The Insights average for the end event is the rate of occurrences over the duration of the unusual activity, between the start Insights event and the end Insights event.

Rate change

The difference between the value of **Baseline average** and **Insight average**, measured as a percentage. For example, if the baseline average of an `AccessDenied` error occurring is 1.0, and the Insight average is 3.0, the rate change is 300%. A rate change for an Insight average that exceeds a baseline average shows an up-arrow next to the value. If the Insights event was logged because the activity is below the baseline average, **Rate change** shows a down-arrow next to the percentage.

If there are no events logged for the attribute or time that you choose, the results list is empty. You can apply only one attribute filter in addition to the time range. If you choose a different attribute filter, your specified time range is preserved.

The following steps describe how to filter by attribute.

To filter by attribute

1. To filter the results by an attribute, choose a lookup attribute from the drop-down menu, and then type or choose a value in the **Enter a lookup value** box.
2. To remove an attribute filter, choose the **X** on the right of the attribute filter box.

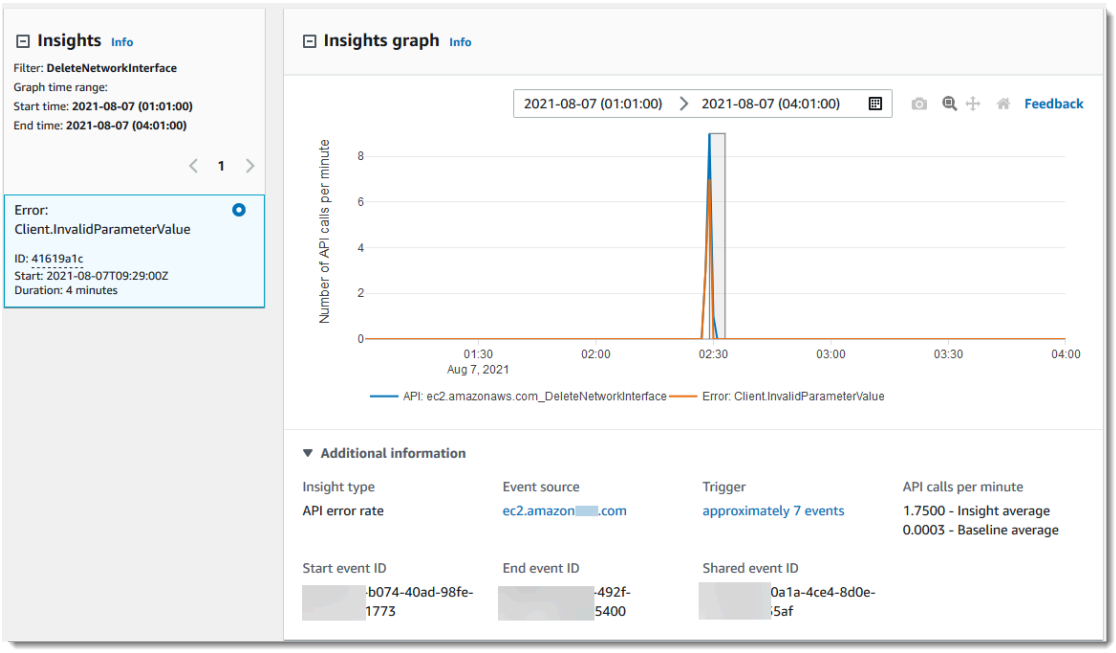
The following steps describe how to filter by a start and end date and time.

To filter by a start and end date and time

1. To narrow the time range for the events that you want to see, choose a time range on the time span bar at the top of the table. Preset time ranges include 30 minutes, 1 hour, 3 hours, or 12 hours. To specify a custom time range, choose **Custom**.
2. Choose one of the following tabs.
   - **Absolute** - Lets you choose a specific time. Go on to the next step.
   - **Relative to selected event** - Selected by default. Lets you choose a time period relative to the start time of an Insights event. Go on to step 4.
3. To set an **Absolute** time range, do the following.
   a. On the **Absolute** tab, choose the day that you want the time range to start. Enter a start time on the selected day. To enter a date manually, type the date in the format `yyyy/mm/dd`. The start and end times use a 24-hour clock, and values must be in the format `hh:mm:ss`. For example, to indicate a 6:30 p.m. start time, enter **`18:30:00`**.
   b. Choose an end date for the range on the calendar, or specify an end date and time below the calendar. Choose **Apply**.
4. To set a **Relative to selected event** time range, do the following.
   a. Choose a preset time period relative to the start time of Insights events. Preset values are available in minutes, hours, days, or weeks. The maximum relative time period is 12 weeks.
   b. If needed, customize the preset value in the boxes below the presets. Choose **Clear** to reset your changes if needed. When you have set the relative time that you want, choose **Apply**.
5. In **To**, choose the day and specify the time that you want to be the end of the time range. Choose **Apply**.
6. To remove a time range filter, choose the calendar icon on the right of the **Time range** box, and then choose **Remove**.

Viewing Insights events details

1. Choose an Insights event in the results list to show its details. The details page for an Insights event shows a graph of the unusual activity timeline.



2. Hover over the highlighted bands to show the start time and duration of each Insights event in the graph.



The following information is shown in the **Additional information** area of the graph:

- **Insight type**. This can be API call rate or API error rate.
- **Trigger**. This is a link to the **Cloudtrail events** tab, which lists the management events that were analyzed to determine that unusual activity occurred.
- **API calls per minute**
  - **Baseline average** - The typical rate of occurrences per minute on the API on which the Insights event was logged, as measured within approximately the preceding seven days, in a specific Region in your account.
  - **Insights average** - The rate of occurrences per minute on this API that triggered the Insights event. The CloudTrail Insights average for the start event is the rate of calls or errors per minute on the API that triggered the Insights event. Typically, this is the first minute of unusual activity. The Insights average for the end event is the rate of API calls or errors per minute over the duration of the unusual activity, between the start Insights event and the end Insights event.
- **Event source**. The Amazon service endpoint on which the unusual number of API calls or errors were logged. In the preceding image, the source is `ec2.amazonaws.com`, which is the service endpoint for Amazon EC2.
- **Event IDs**.
  - **Start event ID** - The ID of the Insights event that was logged at the start of unusual activity.
  - **End event ID** - The ID of the Insights event that was logged at the end of unusual activity.
  - **Shared event ID** - In Insights events, the **Shared event ID** is a GUID that is generated by CloudTrail Insights to uniquely identify a start and end pair of Insights events. **Shared event ID** is common between the start and the end Insights event, and helps to create a correlation between both events to uniquely identify unusual activity.

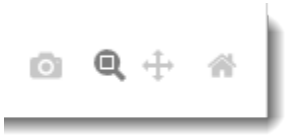
3. Choose the **Attributions** tab to view information about the user identities, user agents, and on API call rate Insights events, error codes correlated with unusual and baseline activity. A maximum of five user identities, five user agents, and five error codes are shown in tables on the **Attributions** tab, sorted by an average of the count of activity, in descending order from highest to lowest. For more information about the **Attributions** tab, see **Attributions** tab (p. 271) and CloudTrail Insights `insightDetails` element (p. 424) in this guide.
4. On the **CloudTrail events** tab, view related events that CloudTrail analyzed to determine that unusual activity occurred. By default, a filter is already applied for the Insights event name, which is also the name of the related API. The **CloudTrail events** tab shows CloudTrail management events related to the subject API that occurred between the start time (minus one minute) and end time (plus one minute) of the Insights event.

   As you select other Insights events in the graph, the events shown in the **CloudTrail events** table change. These events help you perform deeper analysis to determine the probable cause of an Insights event and reasons for unusual API activity.

   To show all CloudTrail events that were logged during the Insights event duration, and not only those for the related API, turn off the filter.
5. Choose the **Insights event record** tab to view the Insights start and end events in JSON format.
6. Choosing the linked **Event source** returns you to the **Insights** page, filtered by that event source.

Zoom, pan, and download graph

You can zoom, pan, and reset the axes of the graph on the Insights event details page by using a toolbar in the upper right corner.

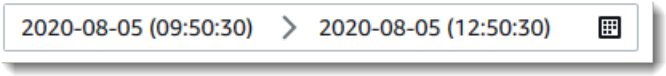


From left to right, the command buttons on the graph toolbar do the following:

- **Download plot as a PNG** - Download the graph image shown on the details page, and save it in PNG format.
- **Zoom** - Drag to select an area on the graph that you want to enlarge and see in greater detail.
- **Pan** - Shift the graph to see adjacent dates or times.
- **Reset axes** - Change graph axes back to the original, clearing zoom and pan settings.

Change graph time span settings

You can change the time span—the selected duration of the events shown on the *x* axis—that is shown in the graph by choosing a setting in the graph's upper right corner.

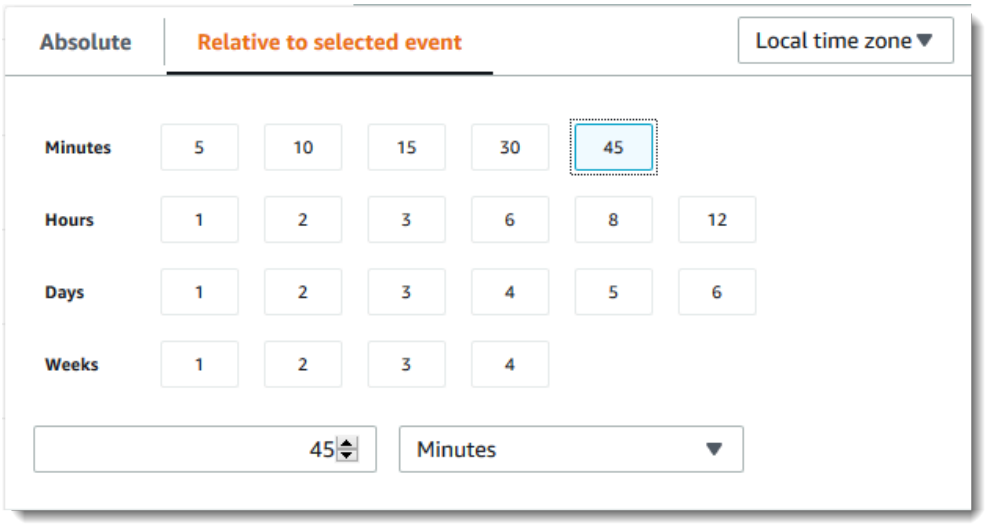


The default time span that is shown in the graph depends on the duration of the selected Insights event.

Duration of Insights event	Default time span
Less than 4 hours	**3h** (three hours)

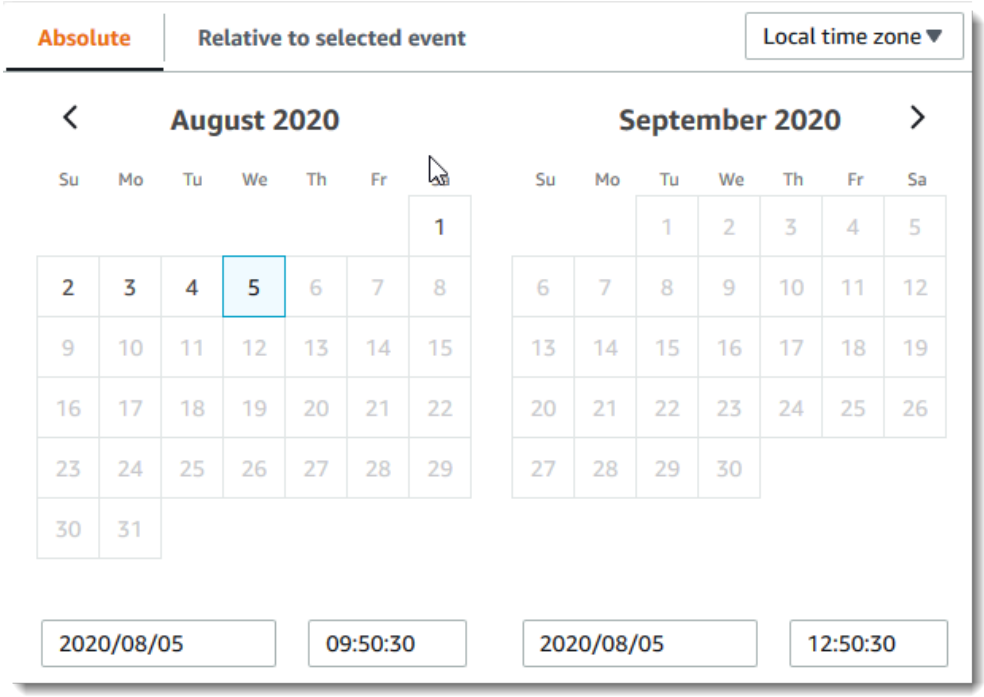
Duration of Insights event	Default time span
Between 4 and 12 hours	**12h**(12 hours)
Between 12 and 24 hours	**1d** (one day)
Between 24 and 72 hours	**3d** (three days)
Longer than 72 hours	**1w** (one week)

You can choose presets of five minutes, 30 minutes, one hour, three hours, 12 hours, or **Custom**. The following image shows **Relative to selected event** time periods you can choose in **Custom** settings. Relative time periods are approximate time periods surrounding the start and end of the selected Insights event that is displayed on an Insights event details page.



To customize a selected preset, specify a number and time unit in the boxes below the presets.

To specify an exact date and time range, choose the **Absolute** tab. If you set an absolute date and time range, start and end times are required. For information about how to set the time, see the section called "Filtering Insights events" (p. 63) in this topic.



Downloading Insights events

You can download recorded Insights event history as a file in CSV or JSON format. Use filters and time ranges to reduce the size of the file you download.

> **Note**
> CloudTrail event history files are data files that contain information (such as resource names) that can be configured by individual users. Some data can potentially be interpreted as commands in programs used to read and analyze this data (CSV injection). For example, when CloudTrail events are exported to CSV and imported to a spreadsheet program, that program might warn you about security concerns. As a security best practice, disable links or macros from downloaded event history files.

1. Specify the filter and time range for events you want to download. For example, you can specify the event name, `StartInstances`, and specify a time range for the last three days of activity.
2. Choose **Download events**, and then choose **Download CSV** or **Download JSON**. You are prompted to choose a location to save the file.

   > **Note**
   > Your download might take some time to finish. For faster results, before you start the download process, use a more specific filter or a shorter time range to narrow the results.
3. After your download is complete, open the file to view the events that you specified.
4. To cancel your download, choose **Cancel download**. If you cancel a download before it is finished, a CSV or JSON file on your local computer might contain only part of your events.

Viewing CloudTrail Insights events with the Amazon CLI

You can look up CloudTrail Insights events for the last 90 days by running the **aws cloudtrail lookup-events** command. The `lookup-events` command has the following options:

- `--end-time`
- `--event-category`
- `--max-results`
- `--start-time`
- `--lookup-attributes`
- `--next-token`
- `--generate-cli-skeleton`
- `--cli-input-json`

For general information about using the Amazon Command Line Interface, see the Amazon Command Line Interface User Guide.

Contents

- Prerequisites (p. 70)
- Getting command line help (p. 70)
- Looking up Insights events (p. 70)
- Specifying the number of Insights events to return (p. 73)
- Looking up Insights events by time range (p. 74)
  - Valid <timestamp> formats (p. 74)
- Looking up Insights events by attribute (p. 74)
  - Attribute lookup examples (p. 74)
- Specifying the next page of results (p. 75)
- Getting JSON input from a file (p. 75)
- Lookup output fields (p. 76)

Prerequisites

- To run Amazon CLI commands, you must install the Amazon CLI. For more information, see Installing the Amazon Command Line Interface.
- Make sure your Amazon CLI version is greater than 1.6.6. To verify the CLI version, run **aws --version** on the command line.
- To set the account, Region, and default output format for an Amazon CLI session, use the **aws configure** command. For more information, see Configuring the Amazon Command Line Interface.
- To log Insights events on API call volume, the trail must log `write` management events. To log Insights events on API error rate, the trail must log `read` or `write` management events.

> **Note**
> The CloudTrail Amazon CLI commands are case-sensitive.

Getting command line help

To see the command line help for `lookup-events`, type the following command.

```
aws cloudtrail lookup-events help
```

Looking up Insights events

To see the ten latest Insights events, type the following command.

```
aws cloudtrail lookup-events --event-category insight
```

A returned event looks similar to the following example,

```
{
    "NextToken": "kbOt5LlZe+
+mErCebpy2TgaMgmDvF1kYGFcH64JSjIbZFjsuvrSqg66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXd12juEXAMPLE=",
    "Events": [
        {
            "eventVersion": "1.07",
            "eventTime": "2019-10-15T21:13:00Z",
            "awsRegion": "us-east-1",
            "eventID": "EXAMPLE-9b6f-45f8-bc6b-9b41c052ebc7",
            "eventType": "AwsCloudTrailInsight",
            "recipientAccountId": "123456789012",
            "sharedEventID": "EXAMPLE8-02b2-4e93-9aab-08ed47ea5fd3",
            "insightDetails": {
            "state": "Start",
            "eventSource": "autoscaling.amazonaws.com",
            "eventName": "CompleteLifecycleAction",
            "insightType": "ApiCallRateInsight",
            "insightContext": {
              "statistics": {
                "baseline": {
                  "average": 0.0000882145
                },
                "insight": {
                  "average": 0.6
                },
                "insightDuration": 5,
                "baselineDuration": 11336
              },
              "attributions": [
                {
                  "attribute": "userIdentityArn",
                  "insight": [
                    {
                      "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
                      "average": 0.2
                    },
                    {
                      "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole2",
                      "average": 0.2
                    },
                    {
                      "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole3",
                      "average": 0.2
                    }
                  ],
                  "baseline": [
                    {
                      "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
                      "average": 0.0000882145
                    }
                  ]
                },
                {
                  "attribute": "userAgent",
                  "insight": [
                    {
                      "value": "codedeploy.amazonaws.com",
                      "average": 0.6
                    }
                  ],
```

```
              "baseline": [
                {
                  "value": "codedeploy.amazonaws.com",
                  "average": 0.0000882145
                }
              ]
            },
            {
              "attribute": "errorCode",
              "insight": [
                {
                  "value": "null",
                  "average": 0.6
                }
              ],
              "baseline": [
                {
                  "value": "null",
                  "average": 0.0000882145
                }
              ]
            }
          ]
        }
      },
        "eventCategory": "Insight"
    },
    {
        "eventVersion": "1.07",
        "eventTime": "2019-10-15T21:14:00Z",
        "awsRegion": "us-east-1",
        "eventID": "EXAMPLEc-9eac-4af6-8e07-26a5ae8786a5",
        "eventType": "AwsCloudTrailInsight",
        "recipientAccountId": "123456789012",
        "sharedEventID": "EXAMPLE8-02b2-4e93-9aab-08ed47ea5fd3",
        "insightDetails": {
        "state": "End",
        "eventSource": "autoscaling.amazonaws.com",
        "eventName": "CompleteLifecycleAction",
        "insightType": "ApiCallRateInsight",
        "insightContext": {
          "statistics": {
            "baseline": {
              "average": 0.0000882145
            },
            "insight": {
              "average": 0.6
            },
            "insightDuration": 5,
            "baselineDuration": 11336
          },
          "attributions": [
            {
              "attribute": "userIdentityArn",
              "insight": [
                {
                  "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
                  "average": 0.2
                },
                {
                  "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole2",
                  "average": 0.2
                },
                {
                  "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole3",
                  "average": 0.2
```

```
                    }
                  ],
                  "baseline": [
                    {
                      "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
                      "average": 0.0000882145
                    }
                  ]
                },
                {
                  "attribute": "userAgent",
                  "insight": [
                    {
                      "value": "codedeploy.amazonaws.com",
                      "average": 0.6
                    }
                  ],
                  "baseline": [
                    {
                      "value": "codedeploy.amazonaws.com",
                      "average": 0.0000882145
                    }
                  ]
                },
                {
                  "attribute": "errorCode",
                  "insight": [
                    {
                      "value": "null",
                      "average": 0.6
                    }
                  ],
                  "baseline": [
                    {
                      "value": "null",
                      "average": 0.0000882145
                    }
                  ]
                }
              ]
            }
          },
            "eventCategory": "Insight"
        }
    ]
}
```


For an explanation of the lookup-related fields in the output, see Lookup output fields (p. 76) in this topic. For an explanation of fields in the Insights event, see CloudTrail record contents (p. 402).

Specifying the number of Insights events to return

To specify the number of events to return, type the following command.

```
aws cloudtrail lookup-events --event-category insight --max-results <integer>
```

The default value for *<integer>*, if it is not specified, is 10. Possible values are 1 through 50. The following example returns one result.

```
aws cloudtrail lookup-events --event-category insight --max-results 1
```

Looking up Insights events by time range

Insights events from the past 90 days are available for lookup. To specify a time range, type the following command.

```
aws cloudtrail lookup-events --event-category insight --start-time <timestamp> --end-
time <timestamp>
```

`--start-time` *<timestamp>* specifies that only Insights events that occur after or at the specified time are returned. If the specified start time is after the specified end time, an error is returned.

`--end-time` *<timestamp>* specifies that only Insights events that occur before or at the specified time are returned. If the specified end time is before the specified start time, an error is returned.

The default start time is the earliest date that data is available within the last 90 days.The default end time is the time of the event that occurred closest to the current time.

Valid *<timestamp>* formats

The `--start-time` and `--end-time` attributes take UNIX time values or valid equivalents.

The following are examples of valid formats. Date, month, and year values can be separated by hyphens or forward slashes. Double quotes must be used if spaces are present.

```
1422317782
1422317782.0
01-27-2015
01-27-2015,01:16PM
"01-27-2015, 01:16 PM"
"01/27/2015, 13:16"
2015-01-27
"2015-01-27, 01:16 PM"
```

Looking up Insights events by attribute

To filter by an attribute, type the following command.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=<attribute>,AttributeValue=<string>
```

You can specify only one attribute key-value pair for each **lookup-events** command. The following are valid Insights event values for `AttributeKey`. Value names are case sensitive.

- `EventId`
- `EventName`
- `EventSource`

Attribute lookup examples

The following example command returns Insights events in which the value of `EventName` is `PutRule`.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=EventName, AttributeValue=PutRule
```

The following example command returns Insights events in which the value of `EventId` is `b5cc8c40-12ba-4d08-a8d9-2bceb9a3e002`.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=EventId, AttributeValue=b5cc8c40-12ba-4d08-a8d9-2bceb9a3e002
```

The following example command returns Insights events in which the value of `EventSource` is `iam.amazonaws.com`.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=EventSource, AttributeValue=iam.amazonaws.com
```

Specifying the next page of results

To get the next page of results from a `lookup-events` command, type the following command.

```
aws cloudtrail lookup-events --event-category insight <same parameters as previous command>
 --next-token=<token>
```

In this command, the value for *<token>* is taken from the first field of the output of the previous command.

When you use `--next-token` in a command, you must use the same parameters as in the previous command. For example, suppose you run the following command.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=EventName, AttributeValue=PutRule
```

To get the next page of results, your next command would look like the following.

```
aws cloudtrail lookup-events --event-category insight --lookup-attributes
 AttributeKey=EventName,AttributeValue=PutRule --next-token=EXAMPLEZe+
+mErCebpy2TgaMgmDvF1kYGFcH64JSjIbZFjsuvrSqg66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXd12juEXAMPLE=
```

Getting JSON input from a file

The Amazon CLI for some Amazon services has two parameters, `--generate-cli-skeleton` and `--cli-input-json`, that you can use to generate a JSON template, which you can modify and use as input to the `--cli-input-json` parameter. This section describes how to use these parameters with `aws cloudtrail lookup-events`. For more information, see Generate CLI Skeleton and CLI Input JSON Parameters.

To look up Insights events by getting JSON input from a file

1. Create an input template for use with `lookup-events` by redirecting the `--generate-cli-skeleton` output to a file, as in the following example.

   ```
   aws cloudtrail lookup-events --event-category insight --generate-cli-skeleton >
    LookupEvents.txt
   ```

   The template file generated (in this case, LookupEvents.txt) looks like the following.

   ```
   {
       "LookupAttributes": [
           {
               "AttributeKey": "",
               "AttributeValue": ""
   ```

```
        }
    ],
    "StartTime": null,
    "EndTime": null,
    "MaxResults": 0,
    "NextToken": ""
}
```

2. Use a text editor to modify the JSON as needed. The JSON input must contain only values that are specified.

   Important
   All empty or null values must be removed from the template before you can use it.

   The following example specifies a time range and maximum number of results to return.

```
{
    "StartTime": "2015-01-01",
    "EndTime": "2015-01-27",
    "MaxResults": 2
}
```

3. To use the edited file as input, use the syntax `--cli-input-json file://<filename>`, as in the following example.

```
aws cloudtrail lookup-events --event-category insight --cli-input-json file://
LookupEvents.txt
```

   Note
   You can use other arguments on the same command line as `--cli-input-json`.

Lookup output fields

Events

A list of lookup events based on the lookup attribute and time range that were specified. The events list is sorted by time, with the latest event listed first. Each entry contains information about the lookup request and includes a string representation of the CloudTrail event that was retrieved.

The following entries describe the fields in each lookup event.

CloudTrailEvent

A JSON string that contains an object representation of the event returned. For information about each of the elements returned, see Record Body Contents.

EventId

A string that contains the GUID of the event returned.

EventName

A string that contains the name of the event returned.

EventSource

The Amazon service that the request was made to.

EventTime

The date and time, in UNIX time format, of the event.

Resources

A list of resources referenced by the event that was returned. Each resource entry specifies a resource type and a resource name.

ResourceName

A string that contains the name of the resource referenced by the event.

ResourceType

A string that contains the type of a resource referenced by the event. When the resource type cannot be determined, null is returned.

Username

A string that contains the user name of the account for the event returned.

NextToken

A string to get the next page of results from a previous `lookup-events` command. To use the token, the parameters must be the same as those in the original command. If no `NextToken` entry appears in the output, there are no more results to return.

For more information about CloudTrail Insights events, see Logging Insights events for trails (p. 267) in this guide.

Creating a trail for your Amazon Web Services account

When you create a trail, you enable ongoing delivery of events as log files to an Amazon S3 bucket that you specify. Creating a trail has many benefits, including:

- A record of events that extends past 90 days.
- The option to automatically monitor and alarm on specified events by sending log events to Amazon CloudWatch Logs.
- The option to query logs and analyze Amazon service activity with Amazon Athena.

Beginning on April 12, 2019, you can view trails only in the Amazon Regions where they log events. If you create a trail that logs events in all Amazon Regions, it appears in the console in all Regions. If you create a trail that only logs events in a single Region, you can view and manage it only in that Region. Creating a multi-Region trail is the default option if you create a trail by using the Amazon CloudTrail console, and is a recommended best practice. To create a single-Region trail, you must use the Amazon CLI.

If you use Amazon Organizations, you can create a trail that will log events for all Amazon accounts in the organization. A trail with the same name will be created in each member account, and events from each trail will be delivered to the Amazon S3 bucket that you specify.

Note
Only the management account or delegated administrator account for an organization can create a trail for the organization. Creating a trail for an organization automatically enables integration between CloudTrail and Organizations. For more information, see Creating a trail for an organization (p. 125).

Topics

- Creating and updating a trail with the console (p. 78)

- Creating, updating, and managing trails with the Amazon Command Line Interface (p. 97)

Creating and updating a trail with the console

You can use the CloudTrail console to create, update, or delete your trails. Trails created using the console are multi-Region. To create a trail that logs events in only one Amazon Web Services Region, use the Amazon CLI (p. 100).

You can create up to five trails for each Region. After you create a trail, CloudTrail automatically starts logging API calls and related events in your account to the Amazon S3 bucket that you specify. To stop logging, you can turn off logging for the trail or delete it.

Using the CloudTrail console to create or update a trail provides the following advantages.

- If this is your first time creating a trail, using the CloudTrail console lets you view the available feature and options.
- If you are configuring a trail to log data events, using the CloudTrail console lets you view the available data types. For more information about logging data events, see Logging data events (p. 242).

For information specific to creating a trail for an organization in Amazon Organizations, see Creating a trail for an organization (p. 125).

Topics

- Creating a trail (p. 78)
- Updating a trail (p. 89)
- Deleting a trail (p. 96)
- Turning off logging for a trail (p. 97)

Creating a trail

Follow the procedure to create a trail that applies to all Regions. A trail that applies to all Regions delivers log files from all Regions to an S3 bucket. After you create the trail, Amazon CloudTrail automatically starts logging the events that you specified.

> **Note**
> After you create a trail, you can configure other Amazon services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see CloudTrail supported services and integrations (p. 24).

Topics

- Creating a trail in the console (basic event selectors) (p. 78)
- Creating a trail in the console (advanced event selectors) (p. 82)
- Next steps (p. 88)

Creating a trail in the console (basic event selectors)

In the console, you create a trail that logs events in all Amazon Regions that you have enabled. This is a recommended best practice. To log events in a single Region (not recommended), use the Amazon CLI (p. 100).

Use the following procedure if you have not yet enabled advanced event selectors. If you have advanced event selectors enabled, see Creating a trail in the console (advanced event selectors) (p. 82) to configure data event logging on your trail.

To create a CloudTrail trail with the Amazon Web Services Management Console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. On the CloudTrail service home page, the **Trails** page, or the **Trails** section of the **Dashboard** page, choose **Create trail**.
3. On the **Create Trail** page, for **Trail name**, type a name for your trail. For more information, see CloudTrail naming requirements (p. 151).
4. If this is an Amazon Organizations organization trail, you can enable the trail for all accounts in your organization. To see this option, you must sign in to the console with a user or role in the management or delegated administrator account. To successfully create an organization trail, be sure that the user or role has sufficient permissions (p. 128). For more information, see Creating a trail for an organization (p. 125).
5. For **Storage location**, choose **Create new S3 bucket** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.

   Note
   If you chose **Use existing S3 bucket**, specify a bucket in **Trail log bucket name**, or choose **Browse** to choose a bucket. The bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail (p. 363).

   To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.
6. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

   If you enable SSE-KMS encryption, choose a **New** or **Existing** Amazon KMS key. In **Amazon KMS Alias**, specify an alias, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Note
   You can also type the ARN of a key from another account. The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).
7. In **Additional settings**, configure the following.

   a. For **Log file validation**, choose **Enabled** to have log digests delivered to your S3 bucket. You can use the digest files to verify that your log files did not change after CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

   b. For **SNS notification delivery**, choose **Enabled** to be notified each time a log is delivered to your bucket. CloudTrail stores multiple events in a log file. SNS notifications are sent for every log file, not for every event. For more information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

      If you enable SNS notifications, for **Create a new SNS topic**, choose **New** to create a topic, or choose **Existing** to use an existing topic. If you are creating a trail that applies to all Regions, SNS notifications for log file deliveries from all Regions are sent to the single SNS topic that you create.

If you choose **New**, CloudTrail specifies a name for the new topic for you, or you can type a name. If you choose **Existing**, choose an SNS topic from the drop-down list. You can also enter the ARN of a topic from another Region or from an account with appropriate permissions. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

If you create a topic, you must subscribe to the topic to be notified of log file delivery. You can subscribe from the Amazon SNS console. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. For more information, see the Amazon Simple Notification Service Getting Started Guide.

8. Optionally, configure CloudTrail to send log files to CloudWatch Logs by choosing **Enabled** in **CloudWatch Logs**. For more information, see Sending events to CloudWatch Logs (p. 277).

   a. If you enable integration with CloudWatch Logs, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name.

   b. If you choose **Existing**, choose a log group from the drop-down list.

   c. Choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

      Note
      When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

9. For **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups and Why use tags for trails? (p. 11).

10. On the **Choose log events** page, choose the event types that you want to log. For **Management events**, do the following.

    a. For **API activity**, choose if you want your trail to log **Read** events, **Write** events, or both. For more information, see Management events (p. 238).

    b. Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail. The default setting is to include all Amazon KMS events.

       The option to log or exclude Amazon KMS events is available only if you log management events on your trail. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

       Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

       To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

11. For **Data events**, you can specify logging data events for Amazon S3 buckets, Amazon Lambda functions, Amazon DynamoDB tables, and many other resource types. By default, trails don't log data events. Additional charges apply for logging data events. For more information, see Data events (p. 243). For CloudTrail pricing, see Amazon CloudTrail Pricing. More data event types

are available if you use advanced event selectors; for more information, see Creating a trail in the console (advanced event selectors) (p. 82) in this topic.

For Amazon S3 buckets:

a. For **Data event source**, choose **S3**.

b. You can choose to log **All current and future S3 buckets**, or you can specify individual buckets or functions. By default, data events are logged for all current and future S3 buckets.

> **Note**
> Keeping the default **All current and future S3 buckets** option enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
> If you are creating a trail for a single Region (done by using the Amazon CLI), choosing **All current and future S3 buckets** enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.

c. If you leave the default, **All current and future S3 buckets**, choose to log **Read** events, **Write** events, or both.

d. To select individual buckets, empty the **Read** and **Write** check boxes for **All current and future S3 buckets**. In **Individual bucket selection**, browse for a bucket on which to log data events. Find specific buckets by typing a bucket prefix for the bucket you want. You can select multiple buckets in this window. Choose **Add bucket** to log data events for more buckets. Choose to log **Read** events, such as `GetObject`, **Write** events, such as `PutObject`, or both.

This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.

To remove a bucket from logging, choose **X**.

12. To add another data type on which to log data events, choose **Add data event type**.
13. For Lambda functions:

a. For **Data event source**, choose **Lambda**.

b. In **Lambda function**, choose **All regions** to log all Lambda functions, or **Input function as ARN** to log data events on a specific function.

To log data events for all Lambda functions in your Amazon account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.

> **Note**
> If you are creating a trail for all Regions, this selection enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
> Logging data events for all functions also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

c. If you choose **Input function as ARN**, enter the ARN of a Lambda function.

Note
If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still select the option to log all functions, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

14. For DynamoDB tables:

    a. For **Data event source**, choose **DynamoDB**.
    b. In **DynamoDB table selection**, choose **Browse** to select a table, or paste in the ARN of a DynamoDB table to which you have access. A DynamoDB table ARN uses the following format:

    ```
    arn:partition:dynamodb:region:account_ID:table/table_name
    ```

    To add another table, choose **Add row**, and browse for a table or paste in the ARN of a table to which you have access.

15. Choose **Insights events** if you want your trail to log CloudTrail Insights events.

    In **Event type**, select **Insights events**. In **Insights events**, choose **API call rate**, **API error rate**, or both. You must be logging **Write** management events to log Insights events for **API call rate**. You must be logging **Read** or **Write** management events to log Insights events for **API error rate**.

    CloudTrail Insights analyzes management events for unusual activity, and logs events when anomalies are detected. By default, trails don't log Insights events. For more information about Insights events, see Logging Insights events for trails (p. 267). Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

    Insights events are delivered to a different folder named `/CloudTrail-Insightof` the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named `S3bucketName/AWSLogs/CloudTrail/`, the S3 bucket name with a new prefix is named `S3bucketName/AWSLogs/CloudTrail-Insight/`.

16. When you are finished choosing event types to log, choose **Next**.
17. On the **Review and create** page, review your choices. Choose **Edit** in a section to change the trail settings shown in that section. When you are ready to create the trail, choose **Create trail**.
18. The new trail appears on the **Trails** page. The **Trails** page shows the trails in your account from all Regions. In about 5 minutes, CloudTrail publishes log files that show the Amazon API calls made in your account. You can see the log files in the S3 bucket that you specified. It can take up to 36 hours for CloudTrail to deliver the first Insights event, if you have enabled Insights event logging, and unusual activity is detected.

Note
CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed.

Creating a trail in the console (advanced event selectors)

In the console, you create a trail that logs events in all Amazon Regions. This is a recommended best practice. To log events in a single Region (not recommended), use the Amazon CLI (p. 100).

Use the following procedure if you have enabled advanced event selectors. If you prefer to use basic data event selectors, see Creating a trail in the console (basic event selectors) (p. 78) to configure data event logging on your trail.

To create a CloudTrail trail with the Amazon Web Services Management Console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. On the CloudTrail service home page, the **Trails** page, or the **Trails** section of the **Dashboard** page, choose **Create trail**.
3. On the **Create Trail** page, for **Trail name**, type a name for your trail. For more information, see CloudTrail naming requirements (p. 151).
4. If this is an Amazon Organizations organization trail, you can enable the trail for all accounts in your organization. To see this option, you must sign in to the console with a user or role in the management or delegated administrator account. To successfully create an organization trail, be sure that the user or role has sufficient permissions (p. 128). For more information, see Creating a trail for an organization (p. 125).
5. For **Storage location**, choose **Create new S3 bucket** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.

   Note
   If you chose **Use existing S3 bucket**, specify a bucket in **Trail log bucket name**, or choose **Browse** to choose a bucket. The bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail (p. 363).

   To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.
6. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

   If you enable SSE-KMS encryption, choose a **New** or **Existing** Amazon KMS key. In **Amazon KMS Alias**, specify an alias, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Note
   You can also type the ARN of a key from another account. For more information, see Updating a resource to use your KMS key (p. 394). The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).
7. In **Additional settings**, configure the following.

   a. For **Log file validation**, choose **Enabled** to have log digests delivered to your S3 bucket. You can use the digest files to verify that your log files did not change after CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

   b. For **SNS notification delivery**, choose **Enabled** to be notified each time a log is delivered to your bucket. CloudTrail stores multiple events in a log file. SNS notifications are sent for every log file, not for every event. For more information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

      If you enable SNS notifications, for **Create a new SNS topic**, choose **New** to create a topic, or choose **Existing** to use an existing topic. If you are creating a trail that applies to all Regions, SNS notifications for log file deliveries from all Regions are sent to the single SNS topic that you create.

If you choose **New**, CloudTrail specifies a name for the new topic for you, or you can type a name. If you choose **Existing**, choose an SNS topic from the drop-down list. You can also enter the ARN of a topic from another Region or from an account with appropriate permissions. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

If you create a topic, you must subscribe to the topic to be notified of log file delivery. You can subscribe from the Amazon SNS console. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. For more information, see the Amazon Simple Notification Service Getting Started Guide.

8. Optionally, configure CloudTrail to send log files to CloudWatch Logs by choosing **Enabled** in **CloudWatch Logs**. For more information, see Sending events to CloudWatch Logs (p. 277).

   a. If you enable integration with CloudWatch Logs, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name.

   b. If you choose **Existing**, choose a log group from the drop-down list.

   c. Choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

      Note
      When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

9. For **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups and Why use tags for trails? (p. 11).

10. On the **Choose log events** page, choose the event types that you want to log. For **Management events**, do the following.

    a. For **API activity**, choose if you want your trail to log **Read** events, **Write** events, or both. For more information, see Management events (p. 238).

    b. Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail. The default setting is to include all Amazon KMS events.

       The option to log or exclude Amazon KMS events is available only if you log management events on your trail. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

       Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

       To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

    c. Choose **Exclude Amazon RDS Data API events** to filter Amazon Relational Database Service Data API events out of your trail. The default setting is to include all Amazon RDS Data API events. For more information about Amazon RDS Data API events, see Logging Data API calls with Amazon CloudTrail in the *Amazon RDS User Guide for Aurora*.

11. To log data events, choose **Data events**.
12. For **Data event type**, choose the resource type on which you want to log data events.

    Note
    To log data events for Amazon Glue tables created by Lake Formation, choose **Lake Formation**.

13. Choose a log selector template. CloudTrail includes predefined templates that log all data events for the resource type. To build a custom log selector template, choose **Custom**.

    Note
    Choosing a predefined template for S3 buckets enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
    If the trail applies only to one Region, choosing a predefined template that logs all S3 buckets enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.
    If you are creating a trail for all Regions, choosing a predefined template for Lambda functions enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
    Logging data events for all functions also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

14. (Optional) In **Selector name**, enter a name to identify your selector. The selector name is a descriptive name for an advanced event selector, such as "Log data events for only two S3 buckets". The selector name is listed as `Name` in the advanced event selector and is viewable if you expand the **JSON view**.
15. In **Advanced event selectors**, build an expression for the specific resources on which you want to log data events. You can skip this step if you are using a predefined log template.

    a. Choose from the following fields. For fields that accept an array (more than one value), CloudTrail adds an OR between values.

    - **`readOnly`** - `readOnly` can be set to **Equals** a value of `true` or `false`. Read-only data events are events that do not change the state of a resource, such as `Get*` or `Describe*` events. Write events add, change, or delete resources, attributes, or artifacts, such as `Put*`, `Delete*`, or `Write*` events. To log both `read` and `write` events, don't add a `readOnly` selector.
    - **`eventName`** - `eventName` can use any operator. You can use it to include or exclude any data event logged to CloudTrail, such as `PutBucket`, `PutItem`, or `GetSnapshotBlock`. You can have multiple values for this field, separated by commas.
    - **`resources.ARN`** - You can use any operator with `resources.ARN`, but if you use **Equals** or **NotEquals**, the value must exactly match the ARN of a valid resource of the type you've specified in the template as the value of `resources.type`.

      The following table shows the valid ARN format for each `resources.Type`.

resources.Type	resources.ARN
AWS::DynamoDB::Table	arn:*partition*:dynamodb:*region*:*account_ID*:table/*table_name*
AWS::Lambda::Function	arn:*partition*:lambda:*region*:*account_ID*:function:*function_nar*
AWS::S3::Object[1]	arn:*partition*:s3:::*bucket_name*/ arn:*partition*:s3:::*bucket_name*/*object_or_file_name*/
AWS::CloudTrail::Channel	arn:*partition*:cloudtrail:*region*:*account_ID*:channel/*channel_UUID*
AWS::CodeWhisperer::Profile	arn:*partition*:codewhisperer:*region*:*account_ID*:profile/*profile_ID*
AWS::Cognito::IdentityPool	arn:*partition*:cognito-identity:*region*:*account_ID*:identitypool/*identity_pool_ID*
AWS::DynamoDB::Stream	arn:*partition*:dynamodb:*region*:*account_ID*:table/*table_name*/stream/*date_time*
AWS::EC2::Snapshot	arn:*partition*:ec2:*region*::snapshot/*snapshot_ID*
AWS::EMRWAL::Workspace	arn:*partition*:emrwal:*region*::workspace/*workspace_name*
AWS::FinSpace::Environment	arn:*partition*:finspace:*region*:*account_ID*:environment/*environment_ID*
AWS::Glue::Table	arn:*partition*:glue:*region*:*account_ID*:table/*database_name*/*table_name*
AWS::GuardDuty::Detector	arn:*partition*:guardduty:*region*:*account_ID*:detector/*detector_ID*
AWS::KendraRanking::ExecutionPlan	arn:*partition*:kendra-ranking:*region*:*account_ID*:rescore-execution-plan/*rescore_execution_plan_ID*
AWS::ManagedBlockchain::Network	arn:*partition*:managedblockchain:::networks/*network_name*

resources.Type	resources.ARN
AWS::ManagedBlockchain::Node	arn:*partition*:managedblockchain:*region*:*account_ID*:nodes/*node_ID*
AWS::MedicalImaging::Datastore	arn:*partition*:medical-imaging:*region*:*account_ID*:datastore/*data_store_ID*
AWS::SageMaker::FeatureGroup	arn:*partition*:sagemaker:*region*:*account_ID*:feature-group/*feature_group_name*
AWS::SageMaker::ExperimentTrialComponent	arn:*partition*:sagemaker:*region*:*account_ID*:experiment-trial-component/*experiment_trial_component_name*
AWS::S3::AccessPoint[2]	arn:*partition*:s3:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3ObjectLambda::AccessPoint	arn:*partition*:s3-object-lambda:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3Outposts::Object	arn:*partition*:s3-outposts:*region*:*account_ID*:*object_path*
AWS::SSMMessages::ControlChannel	arn:*partition*:ssmmessages:*region*:*account_ID*:control-channel/*control_channel_ID*
AWS::VerifiedPermissions::PolicyStore	arn:*partition*:verifiedpermissions:*region*:*account_ID*:policy-store/*policy_store_ID*

[1] To log all data events for all objects in a specific S3 bucket, use the `StartsWith` operator, and include only the bucket ARN as the matching value. The trailing slash is intentional; do not exclude it.

[2] To log events on all objects in an S3 access point, we recommend that you use only the access point ARN, don't include the object path, and use the `StartsWith` or `NotStartsWith` operators.

For more information about the ARN formats of data event resources, see Actions, resources, and condition keys in the *Amazon Identity and Access Management User Guide*.

b. For each field, choose **+ Conditions** to add as many conditions as you need, up to a maximum of 500 specified values for all conditions. For example, to exclude data events for two S3 buckets from data events that are logged on your trail, you can set the field to **resources.ARN**, set the operator for **NotStartsWith**, and then either paste in an S3 bucket ARN, or browse for the S3 buckets for which you do not want to log events.

   To add the second S3 bucket, choose **+ Conditions**, and then repeat the preceding instruction, pasting in the ARN for or browsing for a different bucket.

Note
You can have a maximum of 500 values for all selectors on a trail. This includes arrays of multiple values for a selector such as `eventName`. If you have single values for all selectors, you can have a maximum of 500 conditions added to a selector.
If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still log all functions with a predefined selector template, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

c. Choose **+ Field** to add additional fields as required. To avoid errors, do not set conflicting or duplicate values for fields. For example, do not specify an ARN in one selector to be equal to a value, then specify that the ARN not equal the same value in another selector.

d. Save changes to your custom selector template by choosing **Next**. Do not choose another log selector template, or leave this page, or your custom selectors will be lost.

16. To add another data type on which to log data events, choose **Add data event type**. Repeat steps 12 through this step to configure advanced event selectors for the data event type.

17. Choose **Insights events** if you want your trail to log CloudTrail Insights events.

    In **Event type**, select **Insights events**. You must be logging **Write** management events to log Insights events for **API call rate**. You must be logging **Read** or **Write** management events to log Insights events for **API error rate**.

    CloudTrail Insights analyzes management events for unusual activity, and logs events when anomalies are detected. By default, trails don't log Insights events. For more information about Insights events, see Logging Insights events for trails (p. 267). Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

    Insights events are delivered to a different folder named `/CloudTrail-Insightof` the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named `S3bucketName/AWSLogs/CloudTrail/`, the S3 bucket name with a new prefix is named `S3bucketName/AWSLogs/CloudTrail-Insight/`.

18. When you are finished choosing event types to log, choose **Next**.

19. On the **Review and create** page, review your choices. Choose **Edit** in a section to change the trail settings shown in that section. When you are ready to create the trail, choose **Create trail**.

20. The new trail appears on the **Trails** page. The **Trails** page shows the trails in your account from all Regions. In about 5 minutes, CloudTrail publishes log files that show the Amazon API calls made in your account. You can see the log files in the S3 bucket that you specified. It can take up to 36 hours for CloudTrail to deliver the first Insights event, if you have enabled Insights event logging, and unusual activity is detected.

    Note
    CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed.

Next steps

After you create your trail, you can return to the trail to make changes:

- If you haven't already, you can configure CloudTrail to send log files to CloudWatch Logs. For more information, see Sending events to CloudWatch Logs (p. 277).

- Create a table and use it to run a query in Amazon Athena to analyze your Amazon service activity. For more information, see Creating a Table for CloudTrail Logs in the CloudTrail Console in the Amazon Athena User Guide.
- Add custom tags (key-value pairs) to the trail.
- To create another trail, open the **Trails** page, and choose **Add new trail**.

Updating a trail

This section describes how to change trail settings.

To update a single-Region trail to log events in all Amazon Web Services Regions, or update an multi-Region trail to log events in only a single Region, you must use the Amazon CLI. For more information about how to update a single-Region trail to log events in all Regions, see Converting a trail that applies to one Region to apply to all Regions (p. 101). For more information about how to update an multi-Region trail to log events in a single Region, see Converting a multi-Region trail to a single-Region trail (p. 101).

To update a trail with the Amazon Web Services Management Console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Trails**, and then choose a trail name.
3. In **General details**, choose **Edit** to change the following settings. You cannot change the name of a trail.

   - **Apply trail to my organization** - Change whether this trail is an Amazon Organizations organization trail.
   - **Trail log location** - Change the name of the S3 bucket or prefix in which you are storing logs for this trail.
   - **Log file SSE-KMS encryption** - Choose to enable or disable encrypting log files with SSE-KMS instead of SSE-S3.
   - **Log file validation** - Choose to enable or disable validation of the integrity of log files.
   - **SNS notification delivery** - Choose to enable or disable Amazon Simple Notification Service (Amazon SNS) notifications that log files have been delivered to the bucket specified for the trail.

   a. To change the trail to an Amazon Organizations organization trail, you can choose to enable the trail for all accounts in your organization. For more information, see Creating a trail for an organization (p. 125).

   b. To change the specified bucket in **Storage location**, choose **Create new S3 bucket** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.

      Note
      If you chose **Use existing S3 bucket**, specify a bucket in **Trail log bucket name**, or choose **Browse** to choose a bucket. The bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail (p. 363).

      To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.

   c. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management

Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

If you enable SSE-KMS encryption, choose a **New** or **Existing** Amazon KMS key. In **Amazon KMS Alias**, specify an alias, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

Note
You can also type the ARN of a key from another account. For more information, see Updating a resource to use your KMS key (p. 394). The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).

d. For **Log file validation**, choose **Enabled** to have log digests delivered to your S3 bucket. You can use the digest files to verify that your log files did not change after CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

e. For **SNS notification delivery**, choose **Enabled** to be notified each time a log is delivered to your bucket. CloudTrail stores multiple events in a log file. SNS notifications are sent for every log file, not for every event. For more information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

If you enable SNS notifications, for **Create a new SNS topic**, choose **New** to create a topic, or choose **Existing** to use an existing topic. If you are creating a trail that applies to all Regions, SNS notifications for log file deliveries from all Regions are sent to the single SNS topic that you create.

If you choose **New**, CloudTrail specifies a name for the new topic for you, or you can type a name. If you choose **Existing**, choose an SNS topic from the drop-down list. You can also enter the ARN of a topic from another Region or from an account with appropriate permissions. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

If you create a topic, you must subscribe to the topic to be notified of log file delivery. You can subscribe from the Amazon SNS console. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. For more information, see the Amazon Simple Notification Service Getting Started Guide.

4. In **CloudWatch Logs**, choose **Edit** to change settings for sending CloudTrail log files to CloudWatch Logs. Choose **Enabled** in **CloudWatch Logs** to enable sending log files. For more information, see Sending events to CloudWatch Logs (p. 277).

   a. If you enable integration with CloudWatch Logs, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name.

   b. If you choose **Existing**, choose a log group from the drop-down list.

   c. Choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

   Note

   - When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

- The delegated administrator account cannot currently configure a CloudWatch Logs log group using the console, because the console operation is not supported. The delegated administrator account must use the Amazon CLI or CloudTrail APIs to create an organization trail with a CloudWatch Logs log group.

5. In **Tags**, choose **Edit** to change, add, or delete tags on the trail. Add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups and Why use tags for trails? (p. 11).
6. In **Management events**, choose **Edit** to change management event logging settings.

   a. For **API activity**, choose if you want your trail to log **Read** events, **Write** events, or both. For more information, see Management events (p. 238).

   b. Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail. The default setting is to include all Amazon KMS events.

      The option to log or exclude Amazon KMS events is available only if you log management events on your trail. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

      Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

      To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

7. **Important**
   Steps 7-9 are for configuring data events on your trail if you are using basic event selectors. If you are using advanced event selectors, see Updating data event settings with advanced event selectors (p. 93), then return to this procedure for Step 10 and forward.

   In **Data events**, choose **Edit** to change data event logging settings. With basic event selectors, you can specify logging data events for Amazon S3 buckets, Amazon Lambda functions, DynamoDBtables, or a combination of those resources. Additional data event types are supported with advanced event selectors (p. 93). By default, trails don't log data events. Additional charges apply for logging data events. For more information, see Data events (p. 243). For CloudTrail pricing, see Amazon CloudTrail Pricing.

   For Amazon S3 buckets:

   a. For **Data event source**, choose **S3**.

   b. You can choose to log **All current and future S3 buckets**, or you can specify individual buckets or functions. By default, data events are logged for all current and future S3 buckets.

      Note
      Keeping the default **All current and future S3 buckets** option enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
      If the trail applies only to one Region, choosing **All current and future S3 buckets** enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.

   c. If you leave the default, **All current and future S3 buckets**, choose to log **Read** events, **Write** events, or both.

d. To select individual buckets, empty the **Read** and **Write** check boxes for **All current and future S3 buckets**. In **Individual bucket selection**, browse for a bucket on which to log data events. To find specific buckets, type a bucket prefix for the bucket you want. You can select multiple buckets in this window. Choose **Add bucket** to log data events for more buckets. Choose to log **Read** events, such as `GetObject`, **Write** events, such as `PutObject`, or both.

   This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.

   To remove a bucket from logging, choose **X**.

8. To add another data type on which to log data events, choose **Add data event type**.
9. For Lambda functions:

   a. For **Data event source**, choose **Lambda**.

   b. In **Lambda function**, choose **All regions** to log all Lambda functions, or **Input function as ARN** to log data events on a specific function.

      To log data events for all Lambda functions in your Amazon account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.

      Note
      If you are creating a trail for all Regions, this selection enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
      Logging data events for all functions also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

   c. If you choose **Input function as ARN**, enter the ARN of a Lambda function.

      Note
      If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still select the option to log all functions, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

10. To add another data type on which to log data events, choose **Add data event type**.
11. For DynamoDB tables:

    a. For **Data event source**, choose **DynamoDB**.

    b. In **DynamoDB table selection**, choose **Browse** to select a table, or paste in the ARN of a DynamoDB table to which you have access. A DynamoDB table ARN is in the following format:

    ```
    arn:partition:dynamodb:region:account_ID:table/table_name
    ```

    To add another table, choose **Add row**, and browse for a table or paste in the ARN of a table to which you have access.

12. In **Insights events** choose **Edit** if you want your trail to log CloudTrail Insights events.

In **Event type**, select **Insights events**.

In **Insights events**, choose **API call rate**, **API error rate**, or both. You must be logging **Write** management events to log Insights events for **API call rate**. You must be logging **Read** or **Write** management events to log Insights events for **API error rate**.

CloudTrail Insights analyzes management events for unusual activity, and logs events when anomalies are detected. By default, trails don't log Insights events. For more information about Insights events, see Logging Insights events for trails (p. 267). Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

Insights events are delivered to a different folder named `/CloudTrail-Insight`of the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named `S3bucketName/AWSLogs/CloudTrail/`, the S3 bucket name with a new prefix is named `S3bucketName/AWSLogs/CloudTrail-Insight/`.

13. When you are finished changing settings on your trail, choose **Update trail**.

Updating data event settings with advanced event selectors

1. On the trail's details page, in **Data events**, choose **Edit**.
2. If you are not already logging data events, choose **Data events**.
3. For **Data event type**, choose the resource type on which you want to log data events.

   Note
   To log data events for Amazon Glue tables created by Lake Formation, choose **Lake Formation**.

4. Choose a log selector template. CloudTrail includes predefined templates that log all data events for the resource type. To build a custom log selector template, choose **Custom**.

   Note
   Choosing a predefined template for S3 buckets enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
   If the trail applies only to one Region, choosing a predefined template that logs all S3 buckets enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.
   If you are creating a trail for all Regions, choosing a predefined template for Lambda functions enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
   Logging data events for all functions also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

5. (Optional) In **Selector name**, enter a name to identify your selector. The selector name is a descriptive name for an advanced event selector, such as "Log data events for only two S3 buckets". The selector name is listed as `Name` in the advanced event selector and is viewable if you expand the **JSON view**.

6. In **Advanced event selectors**, build an expression for the specific resources on which you want to collect data events. You can skip this step if you are using a predefined log template.

   a. Choose from the following fields. For fields that accept an array (more than one value), CloudTrail adds an OR between values.

   - **`readOnly`** - `readOnly` can be set to **Equals** a value of `true` or `false`. To log both `read` and `write` events, don't add a `readOnly` selector.
   - **`eventName`** - `eventName` can use any operator. You can use it to include or exclude any data event logged to CloudTrail, such as `PutBucket` or `GetSnapshotBlock`. You can have multiple values for this field, separated by commas.
   - **`resources.ARN`** - You can use any operator with `resources.ARN`, but if you use **Equals** or **NotEquals**, the value must exactly match the ARN of a valid resource of the type you've specified in the template as the value of `resources.type`.

     The following table shows the valid ARN format for each `resources.Type`.

resources.Type	resources.ARN
AWS::DynamoDB::Table	arn:*partition*:dynamodb:*region*:*account_ID*:table/*table_name*
AWS::Lambda::Function	arn:*partition*:lambda:*region*:*account_ID*:function:*function_nar*
AWS::S3::Object[1]	arn:*partition*:s3:::*bucket_name*/ arn:*partition*:s3:::*bucket_name*/*object_or_file_name*/
AWS::CloudTrail::Channel	arn:*partition*:cloudtrail:*region*:*account_ID*:channel/*channel_UUID*
AWS::CodeWhisperer::Profile	arn:*partition*:codewhisperer:*region*:*account_ID*:profile/*profile_ID*
AWS::Cognito::IdentityPool	arn:*partition*:cognito-identity:*region*:*account_ID*:identitypool/*identity_pool_ID*
AWS::DynamoDB::Stream	arn:*partition*:dynamodb:*region*:*account_ID*:table/*table_name*/stream/*date_time*
AWS::EC2::Snapshot	arn:*partition*:ec2:*region*::snapshot/*snapshot_ID*
AWS::EMRWAL::Workspace	arn:*partition*:emrwal:*region*::workspace/*workspace_name*
AWS::FinSpace::Environment	arn:*partition*:finspace:*region*:*account_ID*:environment/*environment_ID*

resources.Type	resources.ARN
AWS::Glue::Table	arn:*partition*:glue:*region*:*account_ID*:table/*database_name*/*table_name*
AWS::GuardDuty::Detector	arn:*partition*:guardduty:*region*:*account_ID*:detector/*detector_ID*
AWS::KendraRanking::ExecutionPlan	arn:*partition*:kendra-ranking:*region*:*account_ID*:rescore-execution-plan/*rescore_execution_plan_ID*
AWS::ManagedBlockchain::Network	arn:*partition*:managedblockchain:::networks/*network_name*
AWS::ManagedBlockchain::Node	arn:*partition*:managedblockchain:*region*:*account_ID*:nodes/*node_ID*
AWS::MedicalImaging::Datastore	arn:*partition*:medical-imaging:*region*:*account_ID*:datastore/*data_store_ID*
AWS::SageMaker::FeatureGroup	arn:*partition*:sagemaker:*region*:*account_ID*:feature-group/*feature_group_name*
AWS::SageMaker::ExperimentTrialComponent	arn:*partition*:sagemaker:*region*:*account_ID*:experiment-trial-component/*experiment_trial_component_name*
AWS::S3::AccessPoint[2]	arn:*partition*:s3:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3ObjectLambda::AccessPoint	arn:*partition*:s3-object-lambda:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3Outposts::Object	arn:*partition*:s3-outposts:*region*:*account_ID*:*object_path*
AWS::SSMMessages::ControlChannel	arn:*partition*:ssmmessages:*region*:*account_ID*:control-channel/*control_channel_ID*
AWS::VerifiedPermissions::PolicyStore	arn:*partition*:verifiedpermissions:*region*:*account_ID*:policy-store/*policy_store_ID*

[1] To log all data events for all objects in a specific S3 bucket, use the `StartsWith` operator, and include only the bucket ARN as the matching value. The trailing slash is intentional; do not exclude it.

[2] To log events on all objects in an S3 access point, we recommend that you use only the access point ARN, don't include the object path, and use the `StartsWith` or `NotStartsWith` operators.

For more information about the ARN formats of data event resources, see Actions, resources, and condition keys in the *Amazon Identity and Access Management User Guide*.

b. For each field, choose **+ Conditions** to add as many conditions as you need, up to a maximum of 500 specified values for all conditions. For example, to exclude data events for two S3 buckets from data events that are logged on your trail, you can set the field to **resources.ARN**, set the operator for **NotStartsWith**, and then either paste in an S3 bucket ARN, or browse for the S3 buckets for which you do not want to log events.

To add the second S3 bucket, choose **+ Conditions**, and then repeat the preceding instruction, pasting in the ARN for or browsing for a different bucket.

Note
You can have a maximum of 500 values for all selectors on a trail. This includes arrays of multiple values for a selector such as `eventName`. If you have single values for all selectors, you can have a maximum of 500 conditions added to a selector.
If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still log all functions with a predefined selector template, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

c. Choose **+ Field** to add additional fields as required. To avoid errors, do not set conflicting or duplicate values for fields. For example, do not specify an ARN in one selector to be equal to a value, then specify that the ARN not equal the same value in another selector.

d. Save changes to your custom selector template by choosing **Next**. Do not choose another log selector template, or leave this page, or your custom selectors will be lost.

7. To add another data type on which to log data events, choose **Add data event type**. Repeat steps 3 through this step to configure advanced event selectors for the data event type.
8. After you choose **Next**, in **Step 2: Choose log events**, review the log selector template options you've chosen. Choose **Edit** to go back and make changes.
9. To configure Insights events and other settings for your trail, go back to the preceding procedure in this topic, Updating a trail (p. 89).

Deleting a trail

You can delete trails with the CloudTrail console. If an organization's management account or delegated administrator account deletes an organization trail, the trail is removed from all member accounts of the organization.

To delete a trail with the CloudTrail console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. Open the **Trails** page of the CloudTrail console.
3. Choose the trail name.
4. At the top of the trail details page, choose **Delete**.

5. When you are prompted to confirm, choose **Delete** to delete the trail permanently. The trail is removed from the list of trails. Log files that were already delivered to the Amazon S3 bucket are not deleted.

   Note

Turning off logging for a trail

When you create a trail, logging is turned on automatically. You can turn off logging for a trail. Existing logs are still stored in the trail's Amazon S3 bucket.

To turn off logging for a trail with the CloudTrail console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Trails**, and then choose the name of the trail.
3. At the top of the trail details page, choose **Stop logging** to turn off logging for the trail.
4. When you are prompted to confirm, choose **Stop logging**. CloudTrail stops logging activity for that trail.
5. To resume logging for that trail, choose **Start logging** on the trail configuration page.

Creating, updating, and managing trails with the Amazon Command Line Interface

You can use the Amazon CLI to create, update, and manage your trails. When using the Amazon CLI, remember that your commands run in the Amazon Region configured for your profile. If you want to run the commands in a different Region, either change the default Region for your profile, or use the **--region** parameter with the command.

Note
You need the Amazon command line tools to run the Amazon Command Line Interface (Amazon CLI) commands in this topic. Make sure you have a recent version of the Amazon CLI installed. For more information, see the Amazon Command Line Interface User Guide. For help with CloudTrail commands at the Amazon CLI command line, type `aws cloudtrail help`.

Commonly used commands for trail creation, management, and status

Some of the more commonly used commands for creating and updating trails in CloudTrail include:

- **create-trail (p. 98)** to create a trail.
- **update-trail (p. 100)** to change the configuration of an existing trail.
- **add-tags (p. 103)** to add one or more tags (key-value pairs) to an existing trail.
- **remove-tags (p. 104)** to remove one or more tags from a trail.
- **list-tags (p. 104)** to return a list of tags associated with a trail.
- **put-event-selectors (p. 107)** to add or modify event selectors for a trail.
- **put-insight-selectors** to add or modify Insights event selectors for an existing trail, and enable or disable Insights events.
- **start-logging (p. 99)** to begin logging events with your trail.
- **stop-logging (p. 125)** to pause logging events with your trail.

- **delete-trail (p. 125)** to delete a trail. This command does not delete the Amazon S3 bucket that contains the log files for that trail, if any.
- **describe-trails (p. 104)** to return information about trails in an Amazon Region.
- **get-trail (p. 104)** to return settings information for a trail.
- **get-trail-status (p. 104)** to return information about the current status of a trail.
- **get-event-selectors (p. 107)** to return information about event selectors configured for a trail.
- **get-insight-selectors** to return information about Insights event selectors configured for a trail.

Supported commands for creating and updating trails: create-trail and update-trail

The `create-trail` and `update-trail` commands offer a variety of functionality for creating and managing trails, including:

- Creating a trail that receives logs across Regions, or update a trail with the `--is-multi-region-trail` option. In most circumstances, you should create trails that log events in all Amazon Regions.
- Creating a trail that receives logs for all Amazon accounts in an organization with the **--is-organization-trail** option.
- Converting a multi-Region trail to single-Region trail with the `--no-is-multi-region-trail` option.
- Enabling or disabling log file encryption with the `--kms-key-id` option. The option specifies an Amazon KMS key that you already created and to which you have attached a policy that allows CloudTrail to encrypt your logs. For more information, see Enabling and disabling CloudTrail log file encryption with the Amazon CLI (p. 396).
- Enabling or disabling log file validation with the `--enable-log-file-validation` and `--no-enable-log-file-validation` options. For more information, see Validating CloudTrail log file integrity (p. 307).
- Specifying a CloudWatch Logs log group and role so that CloudTrail can deliver events to a CloudWatch Logs log group. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

Deprecated commands: create-subscription and update-subscription

Important
The `create-subscription` and `update-subscription` commands were used to create and update trails, but are deprecated. Do not use these commands. They do not provide full functionality for creating and managing trails.
If you configured automation that uses one or both of these commands, we recommend that you update your code or scripts to use supported commands such as **create-trail**.

Using create-trail

You can run the `create-trail` command to create trails that are specifically configured to meet your business needs. When using the Amazon CLI, remember that your commands run in the Amazon Region configured for your profile. If you want to run the commands in a different Region, either change the default Region for your profile, or use the **--region** parameter with the command.

Creating a trail that applies to all Regions

To create a trail that applies to all Regions, use the `--is-multi-region-trail` option. By default, the `create-trail` command creates a trail that logs events only in the Amazon Region where the trail was created. To ensure that you log global service events and capture all management event activity in your Amazon account, you should create trails that log events in all Amazon Regions.

Note
When you create a trail, if you specify an Amazon S3 bucket that was not created with CloudTrail, you need to attach the appropriate policy. See Amazon S3 bucket policy for CloudTrail (p. 363).

The following example creates a trail with the name *my-trail* and a tag with a key named *Group* with a value of *Marketing* that delivers logs from all Regions to an existing bucket named *my-bucket*.

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket --is-multi-region-
trail --tags-list [key=Group,value=Marketing]
```

To confirm that your trail exists in all Regions, the `IsMultiRegionTrail` element in the output shows `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Note
Use the `start-logging` command to start logging for your trail.

Start logging for the trail

After the `create-trail` command completes, run the `start-logging` command to start logging for that trail.

Note
When you create a trail with the CloudTrail console, logging is turned on automatically.

The following example starts logging for a trail.

```
aws cloudtrail start-logging --name my-trail
```

This command doesn't return an output, but you can use the `get-trail-status` command to verify that logging has started.

```
aws cloudtrail get-trail-status --name my-trail
```

To confirm that the trail is logging, the `IsLogging` element in the output shows `true`.

```
{
    "LatestDeliveryTime": 1441139757.497,
    "LatestDeliveryAttemptTime": "2015-09-01T20:35:57Z",
    "LatestNotificationAttemptSucceeded": "2015-09-01T20:35:57Z",
    "LatestDeliveryAttemptSucceeded": "2015-09-01T20:35:57Z",
    "IsLogging": true,
    "TimeLoggingStarted": "2015-09-01T00:54:02Z",
    "StartLoggingTime": 1441068842.76,
    "LatestDigestDeliveryTime": 1441140723.629,
    "LatestNotificationAttemptTime": "2015-09-01T20:35:57Z",
    "TimeLoggingStopped": ""
}
```

Creating a single-Region trail

The following command creates a single-Region trail. The specified Amazon S3 bucket must already exist and have the appropriate CloudTrail permissions applied. For more information, see Amazon S3 bucket policy for CloudTrail (p. 363).

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket
```

For more information, see CloudTrail naming requirements (p. 151).

The following is example output.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Creating a trail that applies to all Regions and that has log file validation enabled

To enable log file validation when using `create-trail`, use the `--enable-log-file-validation` option.

For information about log file validation, see Validating CloudTrail log file integrity (p. 307).

The following example creates a trail that delivers logs from all Regions to the specified bucket. The command uses the `--enable-log-file-validation` option.

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket --is-multi-region-
trail --enable-log-file-validation
```

To confirm that log file validation is enabled, the `LogFileValidationEnabled` element in the output shows `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": true,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Using update-trail

Important
As of November 22, 2021, Amazon CloudTrail makes Amazon CloudFront events available only in the Region where the event was processed, China (Ningxia) Region, cn-northwest-1. For trails monitoring global service events, be sure to convert *single-Region trails* in China (Beijing) Region, cn-north-1, to *multi-Region trails*, to include events from China (Ningxia) Region, cn-northwest-1. For more information about capturing CloudFront events, see Enabling and disabling global service event logging (p. 102) later in this section.

You can use the `update-trail` command to change the configuration settings for a trail. You can also use the **add-tags** and **remove-tags** commands to add and remove tags for a trail. You can only update trails from the Amazon Region where the trail was created (its Home Region). When using the Amazon CLI, remember that your commands run in the Amazon Region configured for your profile. If you want to run the commands in a different Region, either change the default Region for your profile, or use the **--region** parameter with the command.

> **Note**
> If you use the Amazon CLI or one of the Amazon SDKs to modify a trail, be sure that the trail's bucket policy is up-to-date. In order for your bucket to automatically receive events from a new Amazon Web Services Region, the policy must contain the full service name, `cloudtrail.amazonaws.com`. For more information, see Amazon S3 bucket policy for CloudTrail (p. 363).

Topics

- Converting a trail that applies to one Region to apply to all Regions (p. 101)
- Converting a multi-Region trail to a single-Region trail (p. 101)
- Enabling and disabling global service event logging (p. 102)
- Enabling log file validation (p. 102)
- Disabling log file validation (p. 103)

Converting a trail that applies to one Region to apply to all Regions

To change an existing trail so that it applies to all Regions, use the `--is-multi-region-trail` option.

```
aws cloudtrail update-trail --name my-trail --is-multi-region-trail
```

To confirm that the trail now applies to all Regions, the `IsMultiRegionTrail` element in the output shows `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Converting a multi-Region trail to a single-Region trail

To change an existing multi-Region trail so that it applies only to the Region in which it was created, use the `--no-is-multi-region-trail` option.

```
aws cloudtrail update-trail --name my-trail --no-is-multi-region-trail
```

To confirm that the trail now applies to a single Region, the `IsMultiRegionTrail` element in the output shows `false`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
```

```
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Enabling and disabling global service event logging

To change a trail so that it does not log global service events, use the `--no-include-global-service-events` option.

```
aws cloudtrail update-trail --name my-trail --no-include-global-service-events
```

To confirm that the trail no longer logs global service events, the `IncludeGlobalServiceEvents` element in the output shows `false`.

```
{
    "IncludeGlobalServiceEvents": false,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

To change a trail so that it logs global service events, use the `--include-global-service-events` option.

To capture CloudFront events, convert single-Region trails in `cn-north-1` with global service events turned on to multi-Region trails using the following CLI command. Replace *myExistingSingleRegionTrailWithGSE* with the appropriate trail name for your configuration.

```
aws cloudtrail --region cn-north-1 update-trail --name myExistingSingleRegionTrailWithGSE
 --is-multi-region-trail
```

Because global service events are only available in the China (Ningxia) Region beginning November 22, 2021, you can also create a single-Region trail in `cn-northwest-1` to ensure continued logging of CloudFront events.

```
aws cloudtrail --region cn-northwest-1 create-trail --include-global-service-events --
name mySingleRegionTrail --s3-bucket-name myBucket
```

Enabling log file validation

To enable log file validation for a trail, use the `--enable-log-file-validation` option. Digest files are delivered to the Amazon S3 bucket for that trail.

```
aws cloudtrail update-trail --name my-trail --enable-log-file-validation
```

To confirm that log file validation is enabled, the `LogFileValidationEnabled` element in the output shows `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
```

```
    "LogFileValidationEnabled": true,
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Disabling log file validation

To disable log file validation for a trail, use the `--no-enable-log-file-validation` option.

```
aws cloudtrail update-trail --name my-trail-name --no-enable-log-file-validation
```

To confirm that log file validation is disabled, the `LogFileValidationEnabled` element in the output shows `false`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

To validate log files with the Amazon CLI, see Validating CloudTrail log file integrity with the Amazon CLI (p. 309).

Managing trails with the Amazon CLI

The Amazon CLI includes several other commands that help you manage your trails. These commands add tags to trails, get trail status, start and stop logging for trails, and delete a trail. You must run these commands from the same Amazon Region where the trail was created (its Home Region). When using the Amazon CLI, remember that your commands run in the Amazon Region configured for your profile. If you want to run the commands in a different Region, either change the default Region for your profile, or use the **--region** parameter with the command.

Topics

- Add one or more tags to a trail (p. 103)
- List tags for one or more trails (p. 104)
- Remove one or more tags from a trail (p. 104)
- Retrieving trail settings and the status of a trail (p. 104)
- Configuring CloudTrail Insights event selectors (p. 106)
- Configuring event selectors (p. 107)
- Configuring advanced event selectors (p. 111)
- Stopping and starting logging for a trail (p. 125)
- Deleting a trail (p. 125)

Add one or more tags to a trail

To add one or more tags to an existing trail, run the **add-tags** command.

The following example adds a tag with the name *Owner* and the value of *Mary* to a trail with the ARN of *arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail* in the US East (Ohio) Region.

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-east-2:123456789012:trail/my-
trail --tags-list Key=Owner,Value=Mary --region us-east-2
```

If successful, this command returns nothing.

List tags for one or more trails

To view the tags associated with one or more existing trails, use the **list-tags** command.

The following example lists the tags for *Trail1* and *Trail2*.

```
aws cloudtrail list-tags --resource-id-list arn:aws:cloudtrail:us-
east-2:123456789012:trail/Trail1 arn:aws:cloudtrail:us-east-2:123456789012:trail/Trail2
```

If successful, this command returns output similar to the following.

```
{
 "ResourceTagList": [
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-2:123456789012:trail/Trail1",
         "TagsList": [
             {
                 "Value": "Alice",
                 "Key": "Name"
             },
             {
                 "Value": "Ohio",
                 "Key": "Location"
             }
         ]
     },
     {
         "ResourceId": "arn:aws:cloudtrail:us-east-2:123456789012:trail/Trail2",
         "TagsList": [
             {
                 "Value": "Bob",
                 "Key": "Name"
             }
         ]
     }
  ]
}
```

Remove one or more tags from a trail

To remove one or more tags from an existing trail, run the **remove-tags** command.

The following example removes tags with the names *Location* and *Name* from a trail with the ARN of *arn:aws:cloudtrail:us-east-2:123456789012:trail/Trail1* in the US East (Ohio) Region.

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-
east-2:123456789012:trail/Trail1 --tags-list Key=Name Key=Location --region us-east-2
```

If successful, this command returns nothing.

Retrieving trail settings and the status of a trail

Run the `describe-trails` command to retrieve information about trails in an Amazon Region. The following example returns information about trails configured in the US East (Ohio) Region.

```
aws cloudtrail describe-trails --region us-east-2
```

If the command succeeds, you see output similar to the following.

```
{
  "trailList": [
    {
      "Name": "my-trail",
      "S3BucketName": "my-bucket",
      "S3KeyPrefix": "my-prefix",
      "IncludeGlobalServiceEvents": true,
      "IsMultiRegionTrail": true,
      "HomeRegion": "us-east-2"
      "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
      "LogFileValidationEnabled": false,
      "HasCustomEventSelectors": false,
      "SnsTopicName": "my-topic",
      "IsOrganizationTrail": false,
    },
    {
      "Name": "my-special-trail",
      "S3BucketName": "another-bucket",
      "S3KeyPrefix": "example-prefix",
      "IncludeGlobalServiceEvents": false,
      "IsMultiRegionTrail": false,
      "HomeRegion": "us-east-2",
      "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-special-trail",
      "LogFileValidationEnabled": false,
      "HasCustomEventSelectors": true,
      "IsOrganizationTrail": false
    },
    {
      "Name": "my-org-trail",
      "S3BucketName": "my-bucket",
      "S3KeyPrefix": "my-prefix",
      "IncludeGlobalServiceEvents": true,
      "IsMultiRegionTrail": true,
      "HomeRegion": "us-east-1"
      "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-org-trail",
      "LogFileValidationEnabled": false,
      "HasCustomEventSelectors": false,
      "SnsTopicName": "my-topic",
      "IsOrganizationTrail": true
    }
  ]
}
```

Run the `get-trail` command to retrieve settings information about a specific trail. The following example returns settings information for a trail named *my-trail*.

```
aws cloudtrail get-trail - -name my-trail
```

If successful, this command returns output similar to the following.

```
{
   "Trail": {
      "Name": "my-trail",
      "S3BucketName": "my-bucket",
      "S3KeyPrefix": "my-prefix",
      "IncludeGlobalServiceEvents": true,
      "IsMultiRegionTrail": true,
      "HomeRegion": "us-east-2"
```

```
      "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
      "LogFileValidationEnabled": false,
      "HasCustomEventSelectors": false,
      "SnsTopicName": "my-topic",
      "IsOrganizationTrail": false,
   }
}
```

Run the `get-trail-status` command to retrieve the status of a trail. You must either run this command from the Amazon Region where it was created (the Home Region), or you must specify that Region by adding the **--region** parameter.

> **Note**
> If the trail is an organization trail and you are a member account in the organization in Amazon Organizations, you must provide the full ARN of that trail, and not just the name.

```
aws cloudtrail get-trail-status --name my-trail
```

If the command succeeds, you see output similar to the following.

```
{
    "LatestDeliveryTime": 1441139757.497,
    "LatestDeliveryAttemptTime": "2015-09-01T20:35:57Z",
    "LatestNotificationAttemptSucceeded": "2015-09-01T20:35:57Z",
    "LatestDeliveryAttemptSucceeded": "2015-09-01T20:35:57Z",
    "IsLogging": true,
    "TimeLoggingStarted": "2015-09-01T00:54:02Z",
    "StartLoggingTime": 1441068842.76,
    "LatestDigestDeliveryTime": 1441140723.629,
    "LatestNotificationAttemptTime": "2015-09-01T20:35:57Z",
    "TimeLoggingStopped": ""
}
```

In addition to the fields shown in the preceding JSON code, the status contains the following fields if there are Amazon SNS or Amazon S3 errors:

- `LatestNotificationError`. Contains the error emitted by Amazon SNS if a subscription to a topic fails.
- `LatestDeliveryError`. Contains the error emitted by Amazon S3 if CloudTrail cannot deliver a log file to a bucket.

Configuring CloudTrail Insights event selectors

Enable Insights events on a trail by running the **put-insight-selectors**, and specifying `ApiCallRateInsight`, `ApiErrorRateInsight`, or both as the value of the `InsightType` attribute. To view the Insights selector settings for a trail, run the `get-insight-selectors` command. You must either run this command from the Amazon Region where the trail was created (the Home Region), or you must specify that Region by adding the **--region** parameter to the command.

> **Note**
> To log Insights events for `ApiCallRateInsight`, the trail must log `write` management events. To log Insights events for `ApiErrorRateInsight`, the trail must log `read` or `write` management events.

Example trail that logs Insights events

The following example uses **put-insight-selectors** to create an Insights event selector for a trail named *`TrailName3`*. This enables Insights event collection for the *`TrailName3`* trail. The Insights event selector logs both `ApiErrorRateInsight` and `ApiCallRateInsight` Insights event types.

```
aws cloudtrail put-insight-selectors --trail-name TrailName3 --insight-selectors
 '[{"InsightType": "ApiCallRateInsight"},{"InsightType": "ApiErrorRateInsight"}]'
```

The example returns the Insights event selector that is configured for the trail.

```
{
   "InsightSelectors":
      [
         {
            "InsightType": "ApiErrorRateInsight"
         },
         {
            "InsightType": "ApiCallRateInsight"
         }
      ],
   "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName3"
}
```

Example: Turn off collection of Insights events

The following example uses **put-insight-selectors** to remove the Insights event selector for a trail named *TrailName3*. Clearing the JSON string of Insights selectors disables Insights event collection for the *TrailName3* trail.

```
aws cloudtrail put-insight-selectors --trail-name TrailName3 --insight-selectors '[]'
```

The example returns the now-empty Insights event selector that is configured for the trail.

```
{
   "InsightSelectors": [ ],
   "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName3"
}
```

Configuring event selectors

To view the event selector settings for a trail, run the `get-event-selectors` command. You must either run this command from the Amazon Region where it was created (the Home Region), or you must specify that Region by using the **--region** parameter.

```
aws cloudtrail get-event-selectors --trail-name TrailName
```

> **Note**
> If the trail is an organization trail and you are a member account in the organization in Amazon Organizations, you must provide the full ARN of that trail, and not just the name.

The following example returns the default settings for an event selector for a trail.

```
{
    "EventSelectors": [
        {
            "ExcludeManagementEventSources": [],
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
```

```
}
```

To create an event selector, run the `put-event-selectors` command. If you want to log Insights events on the trail, be sure the event selector enables logging of the Insights types you want configured your trail. For more information about logging Insights events, see Logging Insights events for trails (p. 267).

When an event occurs in your account, CloudTrail evaluates the configuration for your trails. If the event matches any event selector for a trail, the trail processes and logs the event. You can configure up to 5 event selectors for a trail and up to 250 data resources for a trail. For more information, see Logging data events (p. 242).

Topics

- Example trail with specific event selectors (p. 108)
- Example trail that logs all management and data events (p. 109)
- Example trail that does not log Amazon Key Management Service events (p. 109)
- Example trail that logs relevant low-volume Amazon Key Management Service events (p. 110)

Example trail with specific event selectors

The following example creates an event selector for a trail named *TrailName* to include read-only and write-only management events, data events for two Amazon S3 bucket/prefix combinations, and data events for a single Amazon Lambda function named *hello-world-python-function*.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-selectors
 '[{"ReadWriteType": "All","IncludeManagementEvents": true,"DataResources":
 [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::mybucket/
prefix","arn:aws:s3:::mybucket2/prefix2"]},{"Type": "AWS::Lambda::Function","Values":
 ["arn:aws:lambda:us-west-2:999999999999:function:hello-world-python-function"]}]}]'
```

The example returns the event selector configured for the trail.

```
{
    "EventSelectors": [
        {
            "ExcludeManagementEventSources": [],
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3:::mybucket/prefix",
                        "arn:aws:s3:::mybucket2/prefix2"
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda:us-west-2:123456789012:function:hello-world-python-
function"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
            ],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

Example trail that logs all management and data events

The following example creates an event selector for a trail named *TrailName2* that includes all events, including read-only and write-only management events, and all data events for all Amazon S3 buckets, Amazon Lambda functions, and Amazon DynamoDB tables in the Amazon account. Because this example uses basic event selectors, it cannot configure logging for S3 events on Amazon Outposts, Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, or other advanced event selector resource types. You must use advanced event selectors to log data events for those resources. For more information, see Configuring advanced event selectors (p. 111).

Note
If the trail applies only to one Region, only events in that Region are logged, even though the event selector parameters specify all Amazon S3 buckets and Lambda functions. Event selectors apply only to the Regions where the trail is created.

```
aws cloudtrail put-event-selectors --trail-name TrailName2 --event-selectors
 '[{"ReadWriteType": "All","IncludeManagementEvents": true,"DataResources":
 [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::"]},{"Type":
 "AWS::Lambda::Function","Values": ["arn:aws:lambda"]},{"Type":
 "AWS::DynamoDB::Table","Values": ["arn:aws:dynamodb"]}]}]'
```

The example returns the event selectors configured for the trail.

```
{
    "EventSelectors": [
        {
            "ExcludeManagementEventSources": [],
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3:::"
                    ],
                    "Type": "AWS::S3::Object"
                },
                {
                    "Values": [
                        "arn:aws:lambda"
                    ],
                    "Type": "AWS::Lambda::Function"
                },
{
                    "Values": [
                        "arn:aws:dynamodb"
                    ],
                    "Type": "AWS::DynamoDB::Table"
                }
            ],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2"
}
```

Example trail that does not log Amazon Key Management Service events

The following example creates an event selector for a trail named *TrailName* to include read-only and write-only management events, but to exclude Amazon Key Management Service (Amazon KMS) events. Because Amazon KMS events are treated as management events, and there can be a high volume of them, they can have a substantial impact on your CloudTrail bill if you have more than one trail that captures management events. The user in this example has chosen to exclude Amazon KMS events from

every trail except for one. To exclude an event source, add `ExcludeManagementEventSources` to your event selectors, and specify an event source in the string value.

If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

To start logging Amazon KMS events to a trail again, pass an empty array as the value of `ExcludeManagementEventSources`.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-
selectors '[{"ReadWriteType": "All","ExcludeManagementEventSources":
 ["kms.amazonaws.com"],"IncludeManagementEvents": true]}]'
```

The example returns the event selector that is configured for the trail.

```
{
    "EventSelectors": [
        {
            "ExcludeManagementEventSources": [ "kms.amazonaws.com" ],
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

To start logging Amazon KMS events to a trail again, pass an empty array as the value of `ExcludeManagementEventSources`, as shown in the following command.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-selectors
 '[{"ReadWriteType": "All","ExcludeManagementEventSources": [],"IncludeManagementEvents":
 true]}]'
```

Example trail that logs relevant low-volume Amazon Key Management Service events

The following example creates an event selector for a trail named *TrailName* to include write-only management events and Amazon KMS events. Because Amazon KMS events are treated as management events, and there can be a high volume of them, they can have a substantial impact on your CloudTrail bill if you have more than one trail that captures management events. The user in this example has chosen to include Amazon KMS **Write** events, which will include `Disable`, `Delete` and `ScheduleKey`, but no longer include high-volume actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` (these are now treated as **Read** events).

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-
selectors '[{"ReadWriteType": "WriteOnly","ExcludeManagementEventSources":
 [],"IncludeManagementEvents": true]}]'
```

The example returns the event selector that is configured for the trail. This logs write-only management events, including Amazon KMS events.

```
{
    "EventSelectors": [
        {
            "ExcludeManagementEventSources": [],
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ReadWriteType": "WriteOnly"
```

```
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

Configuring advanced event selectors

To use advanced event selectors to include or exclude data events instead of basic event selectors, use advanced event selectors on a trail's details page. Advanced event selectors let you log data events on more resource types than basic event selectors. Basic selectors log S3 object activity, Amazon Lambda function execution activity, and DynamoDB tables.

In **Advanced event selectors**, build an expression to collect data events on specific S3 buckets, Amazon Lambda functions, `PutAuditEvents` calls on CloudTrail Lake channels, DynamoDB tables, Amazon S3 on Outposts, Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, S3 Object Lambda access points, Amazon EBS direct APIs on EBS snapshots, S3 access points, DynamoDB streams, Amazon Glue tables created by Lake Formation, Amazon FinSpace environments, Amazon SageMaker metrics experiment trial components, Amazon SageMaker feature stores, Amazon Kendra rescore execution plans, Amazon Cognito identity pools, Amazon GuardDuty detectors, Amazon EMR write-ahead log workspaces, Amazon CodeWhisperer profiles, Amazon Verified Permissions policy stores, Amazon Systems Manager control channels, Amazon Managed Blockchain networks, and Amazon HealthImaging data stores.

For more information advanced event selectors, see Configuring advanced event selectors (p. 111).

To view the advanced event selector settings for a trail, run the following `get-event-selectors` command. You must either run this command from the Amazon Region where the trail was created (the Home Region), or you must specify that Region by adding the **--region** parameter.

```
aws cloudtrail get-event-selectors --trail-name TrailName
```

> **Note**
> If the trail is an organization trail, and you are signed in with a member account in the organization in Amazon Organizations, you must provide the full ARN of the trail, and not just the name.

The following example returns the default settings for advanced event selectors for a trail. By default, no advanced event selectors are configured for a trail.

```
{
    "AdvancedEventSelectors": [],
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

To create an advanced event selector, run the `put-event-selectors` command. When a data event occurs in your account, CloudTrail evaluates the configuration for your trails. If the event matches any advanced event selector for a trail, the trail processes and logs the event. You can configure up to 500 conditions on a trail, including all values specified for all advanced event selectors on your trail. For more information, see Logging data events (p. 242).

Topics

- Example trail with specific advanced event selectors (p. 112)
- Example trail that uses custom advanced event selectors to log all management and data events (p. 114)
- Example trail that uses custom advanced event selectors to log Amazon S3 on Amazon Outposts data events (p. 123)

- Example trail that uses advanced event selectors to exclude Amazon Key Management Service events (p. 124)

Example trail with specific advanced event selectors

The following example creates custom advanced event selectors for a trail named *TrailName* to include read and write management events (by omitting the `readOnly` selector), `PutObject` and `DeleteObject` data events for all Amazon S3 bucket/prefix combinations except for a bucket named `sample_bucket_name` and data events for an Amazon Lambda function named `MyLambdaFunction`. Because these are custom advanced event selectors, each set of selectors has a descriptive name. Note that a trailing slash is part of the ARN value for S3 buckets.

```
aws cloudtrail put-event-selectors --trail-name TrailName --advanced-event-selectors
'[
  {
    "Name": "Log readOnly and writeOnly management events",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Management"] }
    ]
  },
  {
    "Name": "Log PutObject and DeleteObject events for all but one bucket",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3::Object"] },
      { "Field": "eventName", "Equals": ["PutObject","DeleteObject"] },
      { "Field": "resources.ARN", "NotEquals": ["arn:aws:s3:::sample_bucket_name/"] }
    ]
  },
  {
    "Name": "Log data plane actions on MyLambdaFunction",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Lambda::Function"] },
      { "Field": "resources.ARN", "Equals": ["arn:aws:lambda:us-
east-2:111122223333:function/MyLambdaFunction"] }
    ]
  }
]'
```

The example returns the advanced event selectors that are configured for the trail.

```
{
  "AdvancedEventSelectors": [
    {
      "Name": "Log readOnly and writeOnly management events",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [ "Management" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        }
      ]
    },
    {
      "Name": "Log PutObject and DeleteObject events for all but one bucket",
      "FieldSelectors": [
        {
```

```
          "Field": "eventCategory",
          "Equals": [ "Data" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "resources.type",
          "Equals": [ "AWS::S3::Object" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "resources.ARN",
          "Equals": [],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [ "arn:aws:s3:::sample_bucket_name/" ],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
      ]
    },
{
      "Name": "Log data plane actions on MyLambdaFunction",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [ "Data" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "resources.type",
          "Equals": [ "AWS::Lambda::Function" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "eventName",
          "Equals": [ "Invoke" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "resources.ARN",
          "Equals": [ "arn:aws:lambda:us-east-2:111122223333:function/MyLambdaFunction" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
```

```
        }
      ]
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

Example trail that uses custom advanced event selectors to log all management and data events

The following example creates advanced event selectors for a trail named *TrailName2* that includes all events, including read-only and write-only management events, and all data events for all S3 buckets, Amazon Lambda functions, `PutAuditEvents` calls on CloudTrail Lake channels, DynamoDB tables, Amazon S3 on Outposts, Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, S3 Object Lambda access points, Amazon EBS direct APIs on EBS snapshots, S3 access points, DynamoDB streams, Amazon Glue tables created by Lake Formation, Amazon FinSpace environments, Amazon SageMaker metrics experiment trial components, Amazon SageMaker feature stores, Amazon Kendra rescore execution plans, Amazon Cognito identity pools, Amazon GuardDuty detectors, Amazon EMR write-ahead log workspaces, Amazon CodeWhisperer profiles, Amazon Verified Permissions policy stores, Amazon Systems Manager control channels, Amazon Managed Blockchain networks, and Amazon HealthImaging data stores.

Note
If the trail applies only to one Region, only events in that Region are logged, even though the event selector parameters specify all Amazon S3 buckets and Lambda functions. In a single-Region trail, event selectors apply only to the Region where the trail is created.

```
aws cloudtrail put-event-selectors --trail-name TrailName2 \
--advanced-event-selectors '
[
  {
    "Name": "Log readOnly and writeOnly management events",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Management"] }
    ]
  },
  {
    "Name": "Log all events for all Amazon S3 buckets",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3::Object"] }
    ]
  },
  {
    "Name": "Log all events for Lambda functions",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Lambda::Function"] }
    ]
  },
  {
    "Name": "Log all events for DynamoDB tables",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::DynamoDB::Table"] }
    ]
  },
  {
    "Name": "Log all CloudTrail PutAuditEvents activity on a CloudTrail Lake channel",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::CloudTrail::Channel"] }
    ]
  },
```

```
{
  "Name": "Log all events for Amazon S3 on Outposts",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::S3Outposts::Object"] }
  ]
},
{
  "Name": "Log all JSON-RPC calls for Ethereum nodes in Amazon Managed Blockchain",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::ManagedBlockchain::Node"] }
  ]
},
{
  "Name": "Log all events for Amazon S3 Object Lambda access points",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::S3ObjectLambda::AccessPoint"] }
  ]
},
{
  "Name": "Log all Amazon EBS direct API calls on snapshots",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::EC2::Snapshot"] }
  ]
},
{
  "Name": "Log all events for Amazon S3 access points",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::S3::AccessPoint"] }
  ]
},
{
  "Name": "Log all events for DynamoDB streams",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::DynamoDB::Stream"] }
  ]
},
{
  "Name": "Log all events for Amazon Glue tables created by Lake Formation",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::Glue::Table"] }
  ]
},
{
  "Name": "Log all events for FinSpace environments",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::FinSpace::Environment"] }
  ]
},
{
  "Name": "Log all events for SageMaker metrics experiment trial components",
  "FieldSelectors": [
    { "Field": "eventCategory", "Equals": ["Data"] },
    { "Field": "resources.type", "Equals": ["AWS::SageMaker::ExperimentTrialComponent"] }
  ]
},
{
  "Name": "Log all events for SageMaker feature stores",
  "FieldSelectors": [
```

```
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::SageMaker::FeatureGroup"] }
    ]
  },
  {
    "Name": "Log all events for Amazon Kendra Intelligent Ranking rescore execution plans",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::KendraRanking::ExecutionPlan"] }
    ]
  },
  {
    "Name": "Log all events for Amazon Cognito identity pools",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Cognito::IdentityPool"] }
    ]
  },
  {
    "Name": "Log all events for an Amazon GuardDuty detector",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::GuardDuty::Detector"] }
    ]
  },
  {
    ""Name": "Log all events for Amazon EMR write-ahead log workspaces",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::EMRWAL::Workspace"] }
    ]
  },
  {
    "Name": "Log all events for Amazon CodeWhisperer profiles",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::CodeWhisperer::Profile"] }
    ]
  },
  {
    "Name": "Log all events for Amazon Verified Permissions policy stores",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::VerifiedPermissions::PolicyStore"] }
     ]
  },
  {
    "Name": "Log all events for Amazon Systems Manager control channels",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::SSMMessages::ControlChannel"] }
     ]
  },
  {
    "Name": "Log all events for Amazon Managed Blockchain networks",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::ManagedBlockchain::Network"] }
     ]
  },
  {
    "Name": "Log all events for Amazon HealthImaging data stores",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::MedicalImaging::Datastore"] }
     ]
```

```
  }
]'
```

The example returns the advanced event selectors configured for the trail.

```
{
  "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2",
  "AdvancedEventSelectors": [
    {
            "Name": "Log readOnly and writeOnly management events",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Management"
                    ]
                }
            ]
    },
    {
            "Name": "Log all events for all Amazon S3 buckets",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Lambda functions",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Lambda::Function"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for DynamoDB tables",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
```

```
                        "AWS::DynamoDB::Table"
                    ]
                }
            ]
        },
        {
            "Name": "Log all CloudTrail PutAuditEvents activity on a CloudTrail Lake
channel",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::CloudTrail::Channel"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon S3 on Outposts",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3Outposts::Object"
                    ]
                }
            ]
        },
        {
            "Name": "Log all JSON-RPC calls for Ethereum nodes in Amazon Managed
Blockchain",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::ManagedBlockchain::Node"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon S3 Object Lambda access points",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
```

```
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::S3ObjectLambda::AccessPoint"
                ]
            }
        ]
    },
    {
        "Name": "Log all Amazon EBS direct API calls on snapshots",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::EC2::Snapshot"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for Amazon S3 access points",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::S3::AccessPoint"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for DynamoDB streams",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::DynamoDB::Stream"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for Amazon Glue tables created by Lake Formation",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
```

```
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Glue::Table"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for FinSpace environments",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::FinSpace::Environment"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for SageMaker metrics experiment trial components",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SageMaker::ExperimentTrialComponent"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for SageMaker feature stores",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SageMaker::FeatureGroup"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Kendra Intelligent Ranking rescore execution
plans",
            "FieldSelectors": [
```

```
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::KendraRanking::ExecutionPlan"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Cognito identity pools",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Cognito::IdentityPool"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for an Amazon GuardDuty detector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::GuardDuty::Detector"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon EMR write-ahead log workspaces",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::EMRWAL::Workspace"
                    ]
                }
            ]
        },
        {
```

```
            "Name": "Log all events for Amazon CodeWhisperer profiles",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::CodeWhisperer::Profile"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Verified Permissions policy stores",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::VerifiedPermissions::PolicyStore"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Systems Manager control channels",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SSMMessages::ControlChannel"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Managed Blockchain networks",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::ManagedBlockchain::Network"
                    ]
                }
            ]
```

```
        },
        {
            "Name": "Log all events for Amazon HealthImaging data stores",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::MedicalImaging::Datastore"
                    ]
                }
            ]
        }
    ]
}
```

Example trail that uses custom advanced event selectors to log Amazon S3 on Amazon Outposts data events

The following example shows how to configure your trail to include all data events for all Amazon S3 on Amazon Outposts objects in your outpost. In this release, the supported value for S3 on Amazon Outposts events for the `resources.type` field is `AWS::S3Outposts::Object`.

```
aws cloudtrail put-event-selectors --trail-name TrailName --region region \
--advanced-event-selectors \
'[
    {
            "Name": "OutpostsEventSelector",
            "FieldSelectors": [
                { "Field": "eventCategory", "Equals": ["Data"] },
                { "Field": "resources.type", "Equals": ["AWS::S3Outposts::Object"] }
            ]
        }
]'
```

The command returns the following example output.

```
{
    "AdvancedEventSelectors": [
        {
            "Name": "OutpostsEventSelector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3Outposts::Object"
                    ]
                }
            ]
        }
    ],
```

```
  "TrailARN": "arn:aws:cloudtrail:region:123456789012:trail/TrailName"
}
```

Example trail that uses advanced event selectors to exclude Amazon Key Management Service events

The following example creates an advanced event selector for a trail named *TrailName* to include read-only and write-only management events (by omitting the `readOnly` selector), but to exclude Amazon Key Management Service (Amazon KMS) events. Because Amazon KMS events are treated as management events, and there can be a high volume of them, they can have a substantial impact on your CloudTrail bill if you have more than one trail that captures management events. In this release, you can exclude events from `kms.amazonaws.com`.

If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

To start logging Amazon KMS events to a trail again, remove the `eventSource` selector, and run the command again.

```
aws cloudtrail put-event-selectors --trail-name TrailName \
--advanced-event-selectors '
[
  {
    "Name": "Log all management events except KMS events",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Management"] },
      { "Field": "eventSource", "NotEquals": ["kms.amazonaws.com"] }
    ]
  }
]'
```

The example returns the advanced event selectors that are configured for the trail.

```
{
  "AdvancedEventSelectors": [
    {
      "Name": "Log all management events except KMS events",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [ "Management" ],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [],
          "NotStartsWith": [],
          "NotEndsWith": []
        },
        {
          "Field": "eventSource",
          "Equals": [],
          "StartsWith": [],
          "EndsWith": [],
          "NotEquals": [ "kms.amazonaws.com" ],
          "NotStartsWith": [],
          "NotEndsWith": []
        }
      ]
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

To start logging excluded events to a trail again, remove the `eventSource` selector, as shown in the following command.

```
aws cloudtrail put-event-selectors --trail-name TrailName \
--advanced-event-selectors '
[
  {
    "Name": "Log all management events",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Management"] }
    ]
  }
]'
```

Stopping and starting logging for a trail

The following commands start and stop CloudTrail logging.

```
aws cloudtrail start-logging --name awscloudtrail-example
```

```
aws cloudtrail stop-logging --name awscloudtrail-example
```

Note
Before deleting a bucket, run the `stop-logging` command to stop delivering events to the bucket. If you don't stop logging, CloudTrail attempts to deliver log files to a bucket with the same name for a limited period of time.
If you stop logging or delete a trail, CloudTrail Insights is disabled on that trail.

Deleting a trail

You can delete a trail with the following command. You can delete a trail only in the Region it was created (the Home Region).

```
aws cloudtrail delete-trail --name awscloudtrail-example
```

When you delete a trail, you do not delete the Amazon S3 bucket or the Amazon SNS topic associated with it. Use the Amazon Web Services Management Console, Amazon CLI, or service API to delete these resources separately.

Creating a trail for an organization

If you have created an organization in Amazon Organizations, you can create a trail that logs all events for all Amazon accounts in that organization. This is sometimes called an *organization trail*.

The management account for the organization can assign a delegated administrator to manage the organization's trails. For more information on adding a delegated administrator, see Add a CloudTrail delegated administrator (p. 337).

You can choose to edit an existing trail in the management account or delegated administrator account, and apply it to an organization, making it an organization trail. Organization trails log events for the management account and all member accounts in the organization. For more information about Amazon Organizations, see Organizations Terminology and Concepts.

Note
You must sign in with the management account or a delegated administrator account associated with an organization to create an organization trail. You must also have sufficient permissions

for the user or role in the management or delegated administrator account to create the trail. If you don't have sufficient permissions, you won't have the option to apply the trail to an organization.

When you create an organization trail, a trail with the name that you give it is created in every Amazon account that belongs to your organization. Users with CloudTrail permissions in member accounts can see this trail when they log into the Amazon CloudTrail console from their Amazon accounts, or when they run Amazon CLI commands such as `describe-trail`. However, users in member accounts do not have sufficient permissions to delete the organization trail, turn logging on or off, change what types of events are logged, or otherwise change the organization trail in any way.

By default, when you create an organization trail in the CloudTrail console, the trail is a multi-Region trail; that is, it logs events from all Regions in each account in the organization, but only in the Amazon partition in which the trail is created. To log events across all Regions in all Amazon partitions in your organization, create a multi-Region organization trail in each partition.

When you create an organization trail in the console, or when you enable CloudTrail as a trusted service in Organizations, this creates a service-linked role to perform logging tasks in your organization's member accounts. This role is named **AWSServiceRoleForCloudTrail**, and is required for CloudTrail to log events for an organization. If an Amazon account is added to an organization, the organization trail and service-linked role are added to that Amazon account, and logging starts for that account automatically in the organization trail. If an Amazon account is removed from an organization, the organization trail and service-linked role are deleted from the Amazon account that is no longer part of the organization. However, log files for the removed account that were created before the account's removal remain in the Amazon S3 bucket where log files are stored for the trail.

If the management account for an Amazon Organizations organization creates an organization trail, but then is subsequently removed as the organization's management account, any organization trail created using their account becomes a non-organization trail.

In the following example, the organization's management account 111111111111 creates a trail named *`MyOrganizationTrail`* for the organization *`o-exampleorgid`*. The trail logs activity for all accounts in the organization in the same Amazon S3 bucket. All accounts in the organization can see *`MyOrganizationTrail`* in their list of trails, but member accounts cannot remove or modify the organization trail. Only the management account or delegated administrator account can change or delete the trail for the organization. Only the management account can remove a member account from an organization. Similarly, by default, only the management account has access to the Amazon S3 bucket *`my-organization-bucket`* for the trail, and the logs contained within it. The high-level bucket structure for log files contains a folder named with the organization ID, and subfolders named with the account IDs for each account in the organization. Events for each member account are logged in the folder that corresponds to the member account ID. If member account 444444444444 is removed from the organization, *`MyOrganizationTrail`* and the service-linked role no longer appear in Amazon account 444444444444, and no further events are logged for that account by the organization trail. However, the 444444444444 folder remains in the Amazon S3 bucket, with all logs created before the removal of the account from the organization.

In this example, the ARN of the trail created in the management account is `aws:cloudtrail:us-east-2:111111111111:trail/`*`MyOrganizationTrail`*. This ARN is the ARN for the trail in all member accounts as well.

Organization trails are similar to regular trails in many ways. You can create multiple trails for your organization, and choose whether to create an organization trail in all Regions or a single Region, and what kinds of events you want logged in your organization trail, just as in any other trail. However, there are some differences. For example, when you create a trail in the console and choose whether to log data events for Amazon S3 buckets or Amazon Lambda functions, the only resources listed in the CloudTrail console are those for the management account or delegated administrator account, but you can add the ARNs for resources in member accounts. Data events for specified member account resources are logged without having to manually configure cross-account access to those resources. For more information

about logging management events, Insights events, and data events, see Working with CloudTrail log files (p. 236).

> **Note**
> In the console, you create a trail that logs all Regions. This is a recommended best practice; logging activity in all Regions helps you keep your Amazon environment more secure. To create a single-Region trail, use the Amazon CLI (p. 100).

You can also configure other Amazon services to further analyze and act upon the event data collected in CloudTrail logs for an organization trail the same way you would for any other trail. For example, you can analyze the data in an organization trail using Amazon Athena. For more information, see Amazon service integrations with CloudTrail logs (p. 24).

Topics

- Event history and organization trails (p. 127)
- Best practices for moving from member account trails to organization trails (p. 127)
- Prepare for creating a trail for your organization (p. 127)
- Creating a trail for your organization in the console (p. 130)
- Creating a trail for an organization with the Amazon Command Line Interface (p. 135)

Event history and organization trails

When you view events in **Event history** for an organization in Amazon Organizations, you can view the events only for the Amazon Web Services account with which you are signed in. For example, if you are signed in with the organization management account, **Event history** shows the last 90 days of management events for the management account. Organization member account events are not shown in **Event history** for the management account. To view member account events in **Event history**, sign in with the member account.

Best practices for moving from member account trails to organization trails

If you already have CloudTrail trails configured for individual member accounts, but want to move to an organization trail to log events in all accounts, you do not want to lose events by deleting individual member account trails before you create an organization trail. But when you have two trails, you incur higher costs because of the additional copy of events delivered to the organization trail.

To help manage costs, but avoid losing events before log delivery starts on the organization trail, consider keeping both your individual member account trails and your organization trail for up to one day. This ensures that the organization trail logs all events, but you incur duplicate event costs only for one day. After the first day, you can stop logging on (or delete) any individual member account trails.

Prepare for creating a trail for your organization

Before you create a trail for your organization, be sure that your organization management account or delegated administrator account is set up correctly for trail creation.

- Your organization must have all features enabled before you can create a trail for it. For more information, see Enabling All Features in Your Organization.
- The management account or delegated administrator account must have the **AWSServiceRoleForOrganizations** role. This role is created automatically by Organizations when you create your organization, and is required for CloudTrail to log events for an organization. For more information, see Organizations and service-linked roles.

- The user or role that creates the organization trail in the management or delegated administrator account must have sufficient permissions to create an organization trail. You must at least apply either the **AWSCloudTrail_FullAccess** policy, or an equivalent policy, to that role or user. You must also have sufficient permissions in IAM and Organizations to create the service-linked role and enable trusted access. The following example policy shows the minimum required permissions.

  Note
  You shouldn't share the **AWSCloudTrail_FullAccess** policy broadly across your Amazon Web Services account. Instead, you should restrict it to Amazon Web Services account administrators due to the highly sensitive nature of the information collected by CloudTrail. Users with this role have the ability to turn off or reconfigure the most sensitive and important auditing functions in their Amazon Web Services accounts. For this reason, you must closely control and monitor access to this policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "organizations:EnableAWSServiceAccess",
                "organizations:ListAccounts",
                "iam:CreateServiceLinkedRole",
                "organizations:DisableAWSServiceAccess",
                "organizations:DescribeOrganization",
                "organizations:ListAWSServiceAccessForOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

- To use the Amazon CLI or the CloudTrail APIs to create an organization trail, you must enable trusted access for CloudTrail in Organizations, and you must manually create an Amazon S3 bucket with a policy that allows logging for an organization trail. For more information, see Creating a trail for an organization with the Amazon Command Line Interface (p. 135).
- To use an existing IAM role to add monitoring of an organization trail to Amazon CloudWatch Logs, you must manually modify the IAM role to allow delivery of CloudWatch Logs for member accounts to the CloudWatch Logs group for the management or delegated administrator account, as shown in the following example.

  Note
  You must use an IAM role and CloudWatch Logs log group that exists in your own account. You cannot use an IAM role or CloudWatch Logs log group owned by a different account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailCreateLogStream20141101",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:o-exampleorgid_*"
            ]
        },
```

```
        {
            "Sid": "AWSCloudTrailPutLogEvents20141101",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:o-exampleorgid_*"
            ]
        }
    ]
}
```

You can learn more about CloudTrail and Amazon CloudWatch Logs in Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277). In addition, consider the limits on CloudWatch Logs and the pricing considerations for the service before deciding to enable the experience for an organization trail. For more information, see CloudWatch Logs Limits and Amazon CloudWatch Pricing.

- To log data events in your organization trail for specific resources in member accounts, have ready a list of Amazon Resource Names (ARNs) for each of those resources. Member account resources are not displayed in the CloudTrail console when you create a trail; you can browse for resources in the management account on which data event collection is supported, such as S3 buckets. Similarly, if you want to add specific member resources when creating or updating an organization trail at the command line, you need the ARNs for those resources.

  Note
  Additional charges apply for logging data events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

You should also consider reviewing how many trails already exist in the management account, delegated administrator account, and in the member accounts before creating an organization trail. CloudTrail limits the number of trails that can be created in each Region. You cannot exceed this limit in the Region where you create the organization trail in the management account or delegated administrator account. However, the trail will be created in the member accounts even if member accounts have reached the limit of trails in a Region. While the first trail of management events in any Region is free, charges apply to additional trails. To reduce the potential cost of an organization trail, consider deleting any unneeded trails in the management and member accounts. For more information about CloudTrail pricing, see Amazon CloudTrail Pricing.

Security best practices in organization trails

As a security best practice, we recommend adding the `aws:SourceArn` condition key to resource policies (such as those for S3 buckets, KMS keys, or SNS topics) that you use with an organization trail. The value of `aws:SourceArn` is the organization trail ARN (or ARNs, if you are using the same resource for more than one trail, such as the same S3 bucket to store logs for more than one trail). This ensures that the resource, such as an S3 bucket, accepts only data that is associated with the specific trail. The trail ARN must use the account ID of the management account. The following policy snippet shows an example where more than one trail is using the resource.

```
"Condition": {
    "StringEquals": {
      "aws:SourceArn": ["Trail_ARN_1",..., "Trail_ARN_n"]
    }
}
```

For information about how to add condition keys to resource policies, see the following:

- Amazon S3 bucket policy for CloudTrail (p. 363)
- Configure Amazon KMS key policies for CloudTrail (p. 387)
- Amazon SNS topic policy for CloudTrail (p. 369)

Creating a trail for your organization in the console

To create an organization trail from the CloudTrail console, you must sign in to the console as a user or role in the management or delegated administrator account that has sufficient permissions (p. 128). If you don't sign in with the management or delegated administrator account, you won't see the option to apply a trail to an organization when you create or edit a trail from the CloudTrail console.

You can configure an organization trail in multiple ways. For example, you can configure the following details for your organization trail:

- By default, when you create a trail in the console, the trail logs all Amazon Web Services Regions. As a best practice, we strongly recommend logging events in all Regions in your Amazon Web Services account. To create a trail for a single Region, use the Amazon CLI (p. 100). For more information, see How CloudTrail works (p. 1).
- Specify whether to apply the trail to your organization. By default, trails aren't applied to organizations. You must choose this option to create an organization trail.
- Specify which Amazon S3 bucket that receives log files for the organization trail. You can choose an existing Amazon S3 bucket, or create one specifically for the organization trail.
- For management and data events, specify if you want to log **Read** events, **Write** events, or both. CloudTrail Insights (p. 267) events are logged only on management events. You can specify logging data events for resources in the management account by choosing them from the lists in the console, and in member accounts if you specify the ARNs of each resource for which you want to enable data event logging. For more information, see Data events (p. 243).

To create an organization trail with the Amazon Web Services Management Console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.

   You must be signed in using an IAM identity in the management or delegated administrator account with sufficient permissions (p. 128) to create an organization trail.
2. Choose **Trails**, and then choose **Create trail**.
3. On the **Create Trail** page, for **Trail name**, type a name for your trail. For more information, see CloudTrail naming requirements (p. 151).
4. Select **Enable for all accounts in my organization**. You only see this option if you sign in to the console with a user or role in the management or delegated administrator account. To successfully create an organization trail, be sure that the user or role has sufficient permissions (p. 128).
5. For **Storage location**, choose **Create new S3 bucket** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.

   Note
   If you chose **Use existing S3 bucket**, specify a bucket in **Trail log bucket name**, or choose **Browse** to choose a bucket. You can choose a bucket belonging to any account, however, the bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail (p. 363).

   To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.
6. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS

encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

If you enable SSE-KMS encryption, choose a **New** or **Existing** Amazon KMS key. In **Amazon KMS Alias**, specify an alias, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394).

Note
You can also type the ARN of a key from another account. For more information, see Updating a resource to use your KMS key (p. 394). The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).

7. In **Additional settings**, configure the following.

   a. For **Log file validation**, choose **Enabled** to have log digests delivered to your S3 bucket. You can use the digest files to verify that your log files did not change after CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

   b. For **SNS notification delivery**, choose **Enabled** to be notified each time a log is delivered to your bucket. CloudTrail stores multiple events in a log file. SNS notifications are sent for every log file, not for every event. For more information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

      If you enable SNS notifications, for **Create a new SNS topic**, choose **New** to create a topic, or choose **Existing** to use an existing topic. If you are creating a trail that applies to all Regions, SNS notifications for log file deliveries from all Regions are sent to the single SNS topic that you create.

      If you choose **New**, CloudTrail specifies a name for the new topic for you, or you can type a name. If you choose **Existing**, choose an SNS topic from the drop-down list. You can also enter the ARN of a topic from another Region or from an account with appropriate permissions. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

      If you create a topic, you must subscribe to the topic to be notified of log file delivery. You can subscribe from the Amazon SNS console. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. For more information, see the Amazon Simple Notification Service Getting Started Guide.

8. Optionally, configure CloudTrail to send log files to CloudWatch Logs by choosing **Enabled** in **CloudWatch Logs**. For more information, see Sending events to CloudWatch Logs (p. 277).

   a. If you enable integration with CloudWatch Logs, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name.

   b. If you choose **Existing**, choose a log group from the drop-down list.

   c. Choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

      Note

      - When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a

CloudWatch Logs log group, you must choose a log group that exists in your current account.

- The delegated administrator account cannot currently configure a CloudWatch Logs log group using the console, because the console operation is not supported. The delegated administrator account must use the Amazon CLI or CloudTrail APIs to create an organization trail with a CloudWatch Logs log group.

9. For **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups and Why use tags for trails? (p. 11).
10. On the **Choose log events** page, choose the event types that you want to log. For **Management events**, do the following.

    a. For **API activity**, choose if you want your trail to log **Read** events, **Write** events, or both. For more information, see Management events (p. 238).

    b. Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail. The default setting is to include all Amazon KMS events.

    The option to log or exclude Amazon KMS events is available only if you log management events on your trail. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

    Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

    To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

11. For **Data events**, you can specify logging data events for Amazon S3 buckets, Amazon Lambda functions, Amazon DynamoDB tables, or multiple other resource types. By default, trails don't log data events. Additional charges apply for logging data events. For more information, see Data events (p. 243). For CloudTrail pricing, see Amazon CloudTrail Pricing.

    Note
    More data event types are available if you use advanced event selectors. If you have opted in to use advanced event selectors, follow steps in Creating a trail in the console (advanced event selectors) (p. 82) to configure data event logging on your trail.

    For Amazon S3 buckets:

    a. For **Data event source**, choose **S3**.

    b. You can choose to log **All current and future S3 buckets**, or you can specify individual buckets or functions. By default, data events are logged for all current and future S3 buckets.

    Note
    Keeping the default **All current and future S3 buckets** option enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
    If the trail applies only to one Region, selecting the **Select all S3 buckets in your account** option enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.

c. If you leave the default, **All current and future S3 buckets**, choose to log **Read** events, **Write** events, or both.

d. To select individual buckets, empty the **Read** and **Write** check boxes for **All current and future S3 buckets**. In **Individual bucket selection**, browse for a bucket on which to log data events. To find specific buckets, type a bucket prefix for the bucket you want. You can select multiple buckets in this window. Choose **Add bucket** to log data events for more buckets. Choose to log **Read** events, such as `GetObject`, **Write** events, such as `PutObject`, or both.

   This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.

   To remove a bucket from logging, choose **X**.

12. To add another data type on which to log data events, choose **Add data event type**.
13. For Lambda functions:

    a. For **Data event source**, choose **Lambda**.

    b. In **Lambda function**, choose **All regions** to log all Lambda functions, or **Input function as ARN** to log data events on a specific function.

       To log data events for all Lambda functions in your Amazon account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.

       Note
       If you are creating a trail for all Regions, this selection enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
       Logging data events for all functions also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

    c. If you choose **Input function as ARN**, enter the ARN of a Lambda function.

       Note
       If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still select the option to log all functions, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

14. For DynamoDB tables:

    a. For **Data event source**, choose **DynamoDB**.

    b. In **DynamoDB table selection**, choose **Browse** to select a table, or paste in the ARN of a DynamoDB table to which you have access. A DynamoDB table ARN is in the following format:

```
arn:partition:dynamodb:region:account_ID:table/table_name
```

       To add another table, choose **Add row**, and browse for a table or paste in the ARN of a table to which you have access.

15. Choose **Insights events** if you want your trail to log CloudTrail Insights events.

    In **Event type**, select **Insights events**. In **Insights events**, choose **API call rate**, **API error rate**, or both. You must be logging **Write** management events to log Insights events for **API call rate**. You must be logging **Read** or **Write** management events to log Insights events for **API error rate**.

    CloudTrail Insights analyzes management events for unusual activity, and logs events when anomalies are detected. By default, trails don't log Insights events. For more information about Insights events, see Logging Insights events for trails (p. 267). Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

    Insights events are delivered to a different folder named `/CloudTrail-Insight`of the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named `S3bucketName/AWSLogs/CloudTrail/`, the S3 bucket name with a new prefix is named `S3bucketName/AWSLogs/CloudTrail-Insight/`.
16. When you are finished choosing event types to log, choose **Next**.
17. On the **Review and create** page, review your choices. Choose **Edit** in a section to change the trail settings shown in that section. When you are ready to create the trail, choose **Create trail**.
18. The new trail appears on the **Trails** page. An organization trail might take up to 24 hours to be created in all Regions in all member accounts. The **Trails** page shows the trails in your account from all Regions. In about 5 minutes, CloudTrail publishes log files that show the Amazon API calls made in your organization. You can see the log files in the Amazon S3 bucket that you specified.

    Note
    You can't rename a trail after it has been created. Instead, you can delete the trail and create a new one.

Next steps

After you create your trail, you can return to the trail to make changes:

- Change the configuration of your trail by editing it. For more information, see Updating a trail (p. 89).
- If needed, configure the Amazon S3 bucket to allow specific users in member accounts to read the log files for the organization. For more information, see Sharing CloudTrail log files between Amazon accounts (p. 297).
- Configure CloudTrail to send log files to CloudWatch Logs. For more information, see Sending events to CloudWatch Logs (p. 277) and the CloudWatch Logs item (p. 128) in Prepare for creating a trail for your organization (p. 127).
- Create a table and use it to run a query in Amazon Athena to analyze your Amazon service activity. For more information, see Creating a Table for CloudTrail Logs in the CloudTrail Console in the Amazon Athena User Guide.
- Add custom tags (key-value pairs) to the trail.
- To create another organization trail, return to the **Trails** page and choose **Create trail**.

Note
When you configure a trail, you can choose an Amazon S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

Creating a trail for an organization with the Amazon Command Line Interface

You can create an organization trail by using the Amazon CLI. The Amazon CLI is regularly updated with additional functionality and commands. To help ensure success, be sure that you have installed or updated to a recent Amazon CLI version before you begin.

Note
The examples in this section are specific to creating and updating organization trails. For examples of using the Amazon CLI to manage trails, see Managing trails with the Amazon CLI (p. 103) and Configuring CloudWatch Logs monitoring with the Amazon CLI (p. 279).
When creating or updating an organization trail with the Amazon CLI, you must use an Amazon CLI profile in the management account or delegated administrator account with sufficient permissions.
You must configure the Amazon S3 bucket used for an organization trail with sufficient permissions.

Create or update an Amazon S3 bucket to use to store the log files for an organization trail

You must specify an Amazon S3 bucket to receive the log files for an organization trail. This bucket must have a policy that allows CloudTrail to put the log files for the organization into the bucket.

The following is an example policy for an Amazon S3 bucket named *myOrganizationBucket*. This bucket is in an Amazon account with the ID *111111111111*, which is the management account for an organization with the ID *o-exampleorgid* that allows logging for an organization trail. It also allows logging for the *111111111111* account in the event that the trail is changed from an organization trail to a trail for that account only. Replace *myOrganizationBucket*, *111111111111*, *region*, and *trailName* with the appropriate values for your configuration.

The example policy includes an `aws:SourceArn` condition key for the Amazon S3 bucket policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail writes to the S3 bucket only for a specific trail or trails. In an organization trail, the value of `aws:SourceArn` must be a trail ARN that is owned by the management account, and uses the management account ID.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailAclCheck20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::myOrganizationBucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailWrite20150319",
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::myOrganizationBucket/AWSLogs/111111111111/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailOrganizationWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::myOrganizationBucket/AWSLogs/o-exampleorgid/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        }
    ]
}
```

This example policy does not allow any users from member accounts to access the log files created for the organization. By default, organization log files are accessible only to the management account. For information about how to allow read access to the Amazon S3 bucket for IAM users in member accounts, see Sharing CloudTrail log files between Amazon accounts (p. 297).

Enabling CloudTrail as a trusted service in Amazon Organizations

Before you can create an organization trail, you must first enable all features in Organizations. For more information, see Enabling All Features in Your Organization, or run the following command using a profile with sufficient permissions in the management account:

```
aws organizations enable-all-features
```

After you enable all features, you must configure Organizations to trust CloudTrail as a trusted service. .

To create the trusted service relationship between Amazon Organizations and CloudTrail, open a terminal or command line and use a profile in the management account. Run the `aws organizations enable-aws-service-access` command, as demonstrated in the following example.

```
aws organizations enable-aws-service-access --service-principal cloudtrail.amazonaws.com
```

Using create-trail

Creating an organization trail that applies to all Regions

To create an organization trail that applies to all Regions, add the `--is-organization-trail` and `--is-multi-region-trail` options.

> **Note**
> When you create or update an organization trail with the Amazon CLI, you must use an Amazon CLI profile in the management account or delegated administrator account with sufficient permissions.

The following example creates an organization trail that delivers logs from all Regions to an existing bucket named *my-bucket*:

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket --is-organization-
trail --is-multi-region-trail
```

To confirm that your trail exists in all Regions, the `IsOrganizationTrail` and `IsMultiRegionTrail` parameters in the output are both set to `true`:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": true,
    "S3BucketName": "my-bucket"
}
```

> **Note**
> Run the `start-logging` command to start logging for your trail. For more information, see Stopping and starting logging for a trail (p. 125).

Creating an organization trail as a single-Region trail

The following command creates an organization trail that only logs events in a single Amazon Web Services Region, also known as a single-Region trail. The Amazon Region where events are logged is the Region specified in the configuration profile for the Amazon CLI.

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket --is-organization-
trail
```

For more information, see CloudTrail naming requirements (p. 151).

Sample output:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": false,
    "IsOrganizationTrail": true,
    "S3BucketName": "my-bucket"
}
```

By default, the `create-trail` command creates a single-Region trail that does not enable log file validation.

Note
Run the `start-logging` command to start logging for your trail.

Running **update-trail** to update an organization trail

You can run the `update-trail` command to change the configuration settings for an organization trail, or to apply an existing trail for a single Amazon account to an entire organization. Remember that you can run the `update-trail` command only from the Region in which the trail was created.

Note
If you use the Amazon CLI or one of the Amazon SDKs to update a trail, be sure that the trail's bucket policy is up-to-date. For more information, see Creating a trail for an organization with the Amazon Command Line Interface (p. 135).
When you create or update an organization trail with the Amazon CLI, you must use an Amazon CLI profile in the management account or delegated administrator account with sufficient permissions.

Applying an existing trail to an organization

To change an existing trail so that it also applies to an organization instead of a single Amazon account, add the `--is-organization-trail` option, as shown in the following example.

```
aws cloudtrail update-trail --name my-trail --is-organization-trail
```

To confirm that the trail now applies to the organization, the `IsOrganizationTrail` parameter in the output has a value of `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": true,
    "S3BucketName": "my-bucket"
}
```

In the preceding example, the trail was configured to apply to all Regions (`"IsMultiRegionTrail": true`). A trail that applied only to a single Region would show `"IsMultiRegionTrail": false` in the output.

Converting an organization trail that applies to one Region to apply to all Regions

To change an existing organization trail so that it applies to all Regions, add the `--is-multi-region-trail` option as shown in the following example.

```
aws cloudtrail update-trail --name my-trail --is-multi-region-trail
```

To confirm that the trail now applies to all Regions, the `IsMultiRegionTrail` parameter in the output has a value of `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
```

```
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": true,
    "S3BucketName": "my-bucket"
}
```

Copying trail events to CloudTrail Lake

You can copy existing trail events to a CloudTrail Lake event data store to create a point-in-time snapshot of events logged to the trail. Copying trail events does not interfere with the trail's ability to log events and does not modify the trail in any way.

You can copy trail events to an existing event data store configured for CloudTrail events, or you can create a new CloudTrail event data store and choose the **Copy trail events** option as part of event data store creation. For more information about copying trail events to an existing event data store, see Copy trail events to CloudTrail Lake using the CloudTrail console (p. 143). For more information about creating a new event data store, see Create an event data store for CloudTrail events (p. 159).

Copying trail events to a CloudTrail Lake event data store, allows you to run queries on the copied events. CloudTrail Lake queries offer a deeper and more customizable view of events than simple key and value lookups in Event history, or running `LookupEvents`. For more information on CloudTrail Lake, see Working with Amazon CloudTrail Lake (p. 156).

If you are copying trail events to an organization event data store, you must use the management account for the organization. You cannot copy trail events using the delegated administrator account for an organization.

Scenarios

The following table describes some common scenarios for copying trail events and how you accomplish each scenario using the console.

Scenario	How do I accomplish this in the console?
Analyze and query historical trail events in CloudTrail Lake without ingesting new events	Create a new event data store and choose the **Copy trail events** option as part of event data store creation. When creating the event data store, deselect **Ingest events** (step 13 of the procedure) to ensure the event data store contains only the historical events for your trail and no future events.
Replace your existing trail with a CloudTrail Lake event data store	Create an event data store with the same event selectors as your trail to ensure that the event data store has the same coverage as your trail. To avoid duplicating events between the source trail and destination event data store, choose a date range for the copied events that is earlier than the creation of the event data store. After your event data store is created, you can turn off logging for the trail to avoid additional charges. For more information about pricing, see Amazon CloudTrail Pricing.

Topics

- Considerations for copying trail events (p. 140)
- Required permissions for copying trail events (p. 140)
- Copy trail events to CloudTrail Lake using the CloudTrail console (p. 143)

Considerations for copying trail events

Consider the following factors when copying trail events.

- When copying trail events, CloudTrail uses the S3 `GetObject` API operation to retrieve the trail events in the source S3 bucket. There are some S3 archived storage classes, such as S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Outposts, and S3 Intelligent-Tiering Deep Archive tiers that are not accessible by using `GetObject`. To copy trail events stored in these archived storage classes, you must first restore a copy using the S3 `RestoreObject` operation. For information about restoring archived objects, see Restoring Archived Objects in the *Amazon S3 User Guide*.
- When you copy trail events to an event data store, CloudTrail copies all trail events regardless of the configuration of the destination event data store's event types, advanced event selectors, or Amazon Web Services Region.
- Before copying trail events, check the retention period of the event data store. CloudTrail only copies trail events that have an `eventTime` within the event data store's retention period. For example, if an event data store's retention period is 90 days, then CloudTrail will not copy any trail events with an `eventTime` older than 90 days.

  We recommend that when you choose a retention period, you consider both the age of the events that you want to copy as well as how long you want to keep the copied events in your event data store. For example, if you copy trail events that are 5 years old and specify a retention period of 7 years, the event data store will retain those events for two years.
- Before copying trail events, disable any access control lists (ACLs) attached to the source S3 bucket, and update the S3 bucket policy for the destination event data store. For more information about updating the S3 bucket policy, see Amazon S3 bucket policy for copying trail events (p. 142). For more information about disabling ACLs, see Controlling ownership of objects and disabling ACLs for your bucket.
- CloudTrail only copies trail events from Gzip compressed log files that are in the source S3 bucket. CloudTrail does not copy trail events from uncompressed log files, or log files that were compressed using a format other than Gzip.
- To avoid duplicating events between the source trail and destination event data store, choose a time range for the copied events that is earlier than the creation of the event data store.
- By default, CloudTrail only copies CloudTrail events contained in the S3 bucket's `CloudTrail` prefix and the prefixes inside the `CloudTrail` prefix, and does not check prefixes for other Amazon services. If you want to copy CloudTrail events contained in another prefix, you must choose the prefix when you copy trail events.
- To copy trail events to an organization event data store, you must use the management account for the organization. You cannot use the delegated administrator account to copy trail events to an organization event data store.
- If you are copying trail events to an event data store for investigation and do not want to ingest any future events, you can stop ingestion on the event data store when you create it. When creating the event data store, deselect the **Ingest events** option (step 13 of the procedure (p. 160)) to ensure the event data store contains only the historical events for your trail and no future events.

Required permissions for copying trail events

Before copying trail events, ensure you have all the required permissions for your IAM role and the event data store's S3 bucket. You only need to update the IAM role permissions if you choose an existing IAM role to copy trail events. If you choose to create a new IAM role, CloudTrail provides all necessary permissions for the role.

If the source S3 bucket uses a KMS key for data encryption, ensure that the KMS key policy allows CloudTrail to decrypt data in the bucket. If the source S3 bucket uses multiple KMS keys, you must update each key's policy to allow CloudTrail to decrypt the data in the bucket.

Topics

- IAM permissions for copying trail events (p. 141)
- Amazon S3 bucket policy for copying trail events (p. 142)
- KMS key policy for decrypting data in the source S3 bucket (p. 143)

IAM permissions for copying trail events

When copying trail events, you have the option to create a new IAM role, or use an existing IAM role. When you choose a new IAM role, CloudTrail creates an IAM role with the required permissions and no further action is required on your part.

If you choose an existing role, ensure the IAM role's policies allow CloudTrail to copy trail events to the destination event data store. This section provides examples of the required IAM role permission and trust policies.

The following example provides the permissions policy, which allows CloudTrail to copy trail events to the event data store's S3 bucket. Replace *myBucketName*, *myAccountID*, *region*, *prefix*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

Replace *key-region*, *keyAccountID*, and *keyID* with the values for the KMS key used to encrypt the source S3 bucket. You can omit the `AWSCloudTrailImportKeyAccess` statement if the source S3 bucket does not use a KMS key for encryption.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailImportBucketAccess",
      "Effect": "Allow",
      "Action": ["s3:ListBucket", "s3:GetBucketAcl"],
      "Resource": [
        "arn:aws:s3:::myBucketName"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
       }
    },
    {
      "Sid": "AWSCloudTrailImportObjectAccess",
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::myBucketName/prefix",
        "arn:aws:s3:::myBucketName/prefix/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
       }
    },
    {
```

```
      "Sid": "AWSCloudTrailImportKeyAccess",
      "Effect": "Allow",
      "Action": ["kms:GenerateDataKey","kms:Decrypt"],
      "Resource": [
        "arn:aws:kms:key-region:keyAccountID:key/keyID"
      ]
    }
  ]
}
```

The following example provides the IAM trust policy, which allows CloudTrail to assume an IAM role to copy trail events to the event data store's S3 bucket. Replace *myAccountID*, *region*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
      }
    }
  ]
}
```

Amazon S3 bucket policy for copying trail events

By default, Amazon S3 buckets and objects are private. Only the resource owner (the Amazon account that created the bucket) can access the bucket and objects it contains. The resource owner can grant access permissions to other resources and users by writing an access policy.

Before you copy trail events, you must update the S3 bucket policy for the destination event data store to allow CloudTrail to copy trail events to the bucket.

You can add the following statement to the event data store's S3 bucket policy to grant these permissions. Replace *roleArn* and *myBucketName* with the appropriate values for your configuration.

```
{
  "Sid": "AWSCloudTrailImportBucketAccess",
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetBucketAcl",
    "s3:GetObject"
  ],
  "Principal": {
    "AWS": "roleArn"
  },
```

```
  "Resource": [
    "arn:aws:s3:::myBucketName",
    "arn:aws:s3:::myBucketName/*"
  ]
},
```

KMS key policy for decrypting data in the source S3 bucket

If the source S3 bucket uses a KMS key for data encryption, ensure the KMS key policy provides CloudTrail with the `kms:Decrypt` and `kms:GenerateDataKey` permissions required to copy trail events from an S3 bucket with SSE-KMS encryption enabled. If your source S3 bucket uses multiple KMS keys, you must update each key's policy. Updating the KMS key policy allows CloudTrail to decrypt data in the source S3 bucket, run validation checks to ensure that events conform to CloudTrail standards, and copy events into the CloudTrail Lake event data store.

The following example provides the KMS key policy, which allows CloudTrail to decrypt the data in the source S3 bucket. Replace *roleArn*, *myBucketName*, *myAccountID*, *region*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

```
{
  "Sid": "AWSCloudTrailImportDecrypt",
  "Effect": "Allow",
  "Action": [
          "kms:Decrypt",
          "kms:GenerateDataKey"
  ],
  "Principal": {
    "AWS": "roleArn"
  },
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::myBucketName/*"
    },
    "StringEquals": {
      "aws:SourceAccount": "myAccountID",
      "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
    }
  }
}
```

Copy trail events to CloudTrail Lake using the CloudTrail console

Use the following procedure to copy trail events to an existing CloudTrail Lake event data store.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. Choose **Trails** in the left navigation pane of the CloudTrail console.
3. On the **Trails** page, choose the trail, and then choose **Copy events to Lake**. If the source S3 bucket for the trail uses a KMS key for data encryption, ensure that the KMS key policy allows CloudTrail to decrypt data in the bucket. If the source S3 bucket uses multiple KMS keys, you must update each key's policy to allow CloudTrail to decrypt data in the bucket. For more information about updating the KMS key policy, see KMS key policy for decrypting data in the source S3 bucket (p. 143).

4. (Optional) By default, CloudTrail only copies CloudTrail events contained in the S3 bucket's `CloudTrail` prefix and the prefixes inside the `CloudTrail` prefix, and does not check prefixes for other Amazon services. If you want to copy CloudTrail events contained in another prefix, choose **Enter S3 URI**, and then choose **Browse S3** to browse to the prefix.
5. (Optional) For **Event source**, choose the time range for copying the events. If you choose a time range, CloudTrail checks the prefix and log file name to verify the name contains a date between the chosen start and end date before attempting to copy trail events. You can choose a **Relative range** or an **Absolute range**. To avoid duplicating events between the source trail and destination event data store, choose a time range that is earlier than the creation of the event data store.

   - If you choose **Relative range**, you can choose to copy events logged in the last 6 months, 1 year, 2 years, 7 years, or a custom range. CloudTrail copies the events logged within the chosen time period.
   - If you choose **Absolute range**, you can choose a specific start and end date. CloudTrail copies the events that occurred between the chosen start and end dates.
6. For **Delivery location**, choose the destination event data store from the drop-down list. The S3 bucket policy for the event data store must grant CloudTrail access to copy trail events into the bucket. For more information about updating the S3 bucket policy, see Amazon S3 bucket policy for copying trail events (p. 142).
7. For **Permissions**, choose from the following IAM role options. If you choose an existing IAM role, verify that the IAM role policy provides the necessary permissions. For more information about updating the IAM role permissions, see IAM permissions for copying trail events (p. 141).

   - Choose **Create a new role (recommended)** to create a new IAM role. For **Enter IAM role name**, enter a name for the role. CloudTrail automatically creates the necessary permissions for this new role.
   - Choose **Use a custom IAM role** to use a custom IAM role that is not listed. For **Enter IAM role ARN**, enter the IAM ARN.
   - Choose **Use an existing role** to choose an existing IAM role from the drop-down list.
8. Choose **Copy events**.
9. You are prompted to confirm the copy. When you are ready to confirm, choose **Copy trail events to Lake**, and then choose **Copy events**.
10. On the **Copy details** page, you can see the copy status and review any failures. When a trail event copy completes, its **Copy status** is set to either **Completed** if there were no errors, or **Failed** if errors occurred.

    Note
    Details shown on the event copy details page are not in real-time. The actual values for details such as **Prefixes copied** may be higher than what is shown on the page. CloudTrail updates the details incrementally over the course of the event copy.
11. If the **Copy status** is **Failed**, fix any errors shown in **Copy failures**, and then choose **Retry copy**. When you retry a copy, CloudTrail resumes the copy at the location where the failure occurred.

For more information about viewing the details of a trail event copy, see Event copy details (p. 179).

Getting and viewing your CloudTrail log files

After you create a trail and configure it to capture the log files you want, you need to be able to find the log files and interpret the information they contain.

CloudTrail delivers your log files to an Amazon S3 bucket that you specify when you create the trail. CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed. Review the Amazon CloudTrail Service Level Agreement for more information. Insights

events are typically delivered to your bucket within 30 minutes of unusual activity. After you enable Insights events for the first time, allow up to 36 hours to see the first Insights events, if unusual activity is detected.

Topics

- Finding your CloudTrail log files (p. 145)
- Downloading your CloudTrail log files (p. 146)

Finding your CloudTrail log files

CloudTrail publishes log files to your S3 bucket in a gzip archive. In the S3 bucket, the log file has a formatted name that includes the following elements:

- The bucket name that you specified when you created trail (found on the Trails page of the CloudTrail console)
- The (optional) prefix you specified when you created your trail
- The string "AWSLogs"
- The account number
- The string "CloudTrail"
- A Region identifier such as us-west-1
- The year the log file was published in YYYY format
- The month the log file was published in MM format
- The day the log file was published in DD format
- An alphanumeric string that disambiguates the file from others that cover the same time period

The following example shows a complete log file object name:

```
bucket_name/prefix_name/AWSLogs/Account ID/CloudTrail/region/YYYY/MM/DD/file_name.json.gz
```

Note
For organization trails, the log file object name includes the organization unit ID in the path, as follows:

```
bucket_name/prefix_name/AWSLogs/O-ID/Account ID/
CloudTrail/Region/YYYY/MM/DD/file_name.json.gz
```

To retrieve a log file, you can use the Amazon S3 console, the Amazon S3 command line interface (CLI), or the API.

To find your log files with the Amazon S3 console

1. Open the Amazon S3 console.
2. Choose the bucket you specified.
3. Navigate through the object hierarchy until you find the log file you want.

   All log files have a .gz extension.

You will navigate through an object hierarchy that is similar to the following example, but with a different bucket name, account ID, Region, and date.

```
All Buckets
    Bucket_Name
        AWSLogs
            123456789012
                CloudTrail
                    us-west-1
                        2014
                            06
                                20
```

A log file for the preceding object hierarchy will look like the following:

```
123456789012_CloudTrail_us-west-1_20140620T1255ZHdkvFTXOA3Vnhbc.json.gz
```

Note
Although uncommon, you may receive log files that contain one or more duplicate events. Duplicate events will have the same **eventID**. For more information about the **eventID** field, see CloudTrail record contents (p. 402).

Downloading your CloudTrail log files

Log files are in JSON format. If you have a JSON viewer add-on installed, you can view the files directly in your browser. Double-click the log file name in the bucket to open a new browser window or tab. The JSON displays in a readable format.

For example, if you use Mozilla Firefox, you can also download the JSONView add-on. With JSONView, you can double-click the compressed .gz file in your bucket to open the log file in JSON format.

CloudTrail log files are Amazon S3 objects. You can use the Amazon S3 console, the Amazon Command Line Interface (CLI), or the Amazon S3 API to retrieve log files.

For more information, see Working with Amazon S3 Objects in the *Amazon Simple Storage Service User Guide.*

The following procedure describes how to download a log file with the Amazon Web Services Management Console.

To download and read a log file

1. Open the Amazon S3 console at https://console.amazonaws.cn/s3/.
2. Choose the bucket and choose the log file that you want to download.
3. Choose **Download** or **Download as** and follow the prompts to save the file. This saves the file in compressed format.

   Note
   Some browsers, such as Chrome, automatically extract the log file for you. If your browser does this for you, skip to step 5.
4. Use a product such as 7-Zip to extract the log file.
5. Open the log file in a text editor such as Notepad++.

For more information about the event fields that can appear in a log file entry, see CloudTrail log event reference (p. 400).

Amazon partners with third-party specialists in logging and analysis to provide solutions that use CloudTrail output. For more information, see Amazon Partner Network - Amazon CloudTrail Partners.

Note
You can also use the **Event history** feature to look up events for create, update, and delete API activity during the last 90 days.
For more information, see Viewing events with CloudTrail Event history (p. 50).

Configuring Amazon SNS notifications for CloudTrail

You can be notified when CloudTrail publishes new log files to your Amazon S3 bucket. You manage notifications using Amazon Simple Notification Service (Amazon SNS).

Notifications are optional. If you want notifications, you configure CloudTrail to send update information to an Amazon SNS topic whenever a new log file has been sent. To receive these notifications, you can use Amazon SNS to subscribe to the topic. As a subscriber you can get updates sent to a Amazon Simple Queue Service (Amazon SQS) queue, which enables you to handle these notifications programmatically.

Topics

- Configuring CloudTrail to send notifications (p. 147)

Configuring CloudTrail to send notifications

You can configure a trail to use an Amazon SNS topic. You can use the CloudTrail console or the **aws cloudtrail create-trail** CLI command to create the topic. CloudTrail creates the Amazon SNS topic for you and attaches an appropriate policy, so that CloudTrail has permission to publish to that topic.

When you create an SNS topic name, the name must meet the following requirements:

- Between 1 and 256 characters long
- Contain uppercase and lowercase ASCII letters, numbers, underscores, or hyphens

When you configure notifications for a trail that applies to all Regions, notifications from all Regions are sent to the Amazon SNS topic that you specify. If you have one or more Region-specific trails, you must create a separate topic for each Region and subscribe to each individually.

To receive notifications, subscribe to the Amazon SNS topic or topics that CloudTrail uses. You do this with the Amazon SNS console or Amazon SNS CLI commands. For more information, see Subscribe to a topic in the *Amazon Simple Notification Service Developer Guide*.

Note
CloudTrail sends a notification when log files are written to the Amazon S3 bucket. An active account can generate a large number of notifications. If you subscribe with email or SMS, you can receive a large volume of messages. We recommend that you subscribe using Amazon Simple Queue Service (Amazon SQS), which lets you handle notifications programmatically. For more information, see Subscribing a Queue to an Amazon SNS Topic in the *Amazon Simple Queue Service Developer Guide*.

The Amazon SNS notification consists of a JSON object that includes a `Message` field. The `Message` field lists the full path to the log file, as shown in the following example:

```
{
    "s3Bucket": "your-bucket-name","s3ObjectKey": ["AWSLogs/123456789012/
CloudTrail/us-east-2/2013/12/13/123456789012_CloudTrail_us-
west-2_20131213T1920Z_LnPgDQnpkSKEsppV.json.gz"]
```

```
}
```

If multiple log files are delivered to your Amazon S3 bucket, a notification may contain multiple logs, as shown in the following example:

```
{
    "s3Bucket": "your-bucket-name",
    "s3ObjectKey": [
        "AWSLogs/123456789012/CloudTrail/us-east-2/2016/08/11/123456789012_CloudTrail_us-
east-2_20160811T2215Z_kpaMYavMQA9Ahp7L.json.gz",
        "AWSLogs/123456789012/CloudTrail/us-east-2/2016/08/11/123456789012_CloudTrail_us-
east-2_20160811T2210Z_zqDkyQv3TK8ZdLr0.json.gz",
        "AWSLogs/123456789012/CloudTrail/us-east-2/2016/08/11/123456789012_CloudTrail_us-
east-2_20160811T2205Z_jaMVRa6JfdLCJYHP.json.gz"
    ]
}
```

If you choose to receive notifications by email, the body of the email consists of the content of the `Message` field. For a complete description of the JSON structure, see Sending Amazon SNS Messages to Amazon SQS Queues in the *Amazon Simple Notification Service Developer Guide*. Only the `Message` field shows CloudTrail information. The other fields contain information from the Amazon SNS service.

If you create a trail with the CloudTrail API, you can specify an existing Amazon SNS topic that you want CloudTrail to send notifications to with the `CreateTrail` or `UpdateTrail` operations. You must make sure that the topic exists and that it has permissions that allow CloudTrail to send notifications to it. See Amazon SNS topic policy for CloudTrail (p. 369).

Additional resources

For more information about Amazon SNS topics and about subscribing to them, see the *Amazon Simple Notification Service Developer Guide*.

Controlling user permissions for CloudTrail

Amazon CloudTrail integrates with Amazon Identity and Access Management (IAM) to help you to control access to CloudTrail and other Amazon resources that CloudTrail requires. Examples of these resources include Amazon S3 buckets and Amazon Simple Notification Service (Amazon SNS) topics. You can use IAM to control which Amazon users can create, configure, or delete CloudTrail trails, event data stores, or channels, start and stop logging, and access the buckets that contain log information. To learn more, see Identity and Access Management for Amazon CloudTrail (p. 340).

The following topics help you understand permissions, policies, and CloudTrail security:

- Granting permissions for CloudTrail administration (p. 355)
- Amazon S3 bucket naming rules
- Amazon S3 bucket policy for CloudTrail (p. 363)
- Amazon S3 bucket policy for CloudTrail Lake query results (p. 368)
- An example of a bucket policy for an organization trail in Creating a trail for an organization with the Amazon Command Line Interface (p. 135).
- Amazon SNS topic policy for CloudTrail (p. 369)
- Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386)
- Required permissions for copying trail events (p. 140)
- Required permissions to assign a delegated administrator (p. 337)
- Default KMS key policy created in CloudTrail console (p. 394)

- Granting permission to view Amazon Config information on the CloudTrail console (p. 360)
- Sharing CloudTrail log files between Amazon accounts (p. 297)
- Required permissions for creating an organization trail (p. 128)
- Using a previously-existing IAM role to add monitoring of an organization trail to Amazon CloudWatch Logs (p. 128)

Tips for managing trails

- Beginning on April 12, 2019, trails are viewable only in the Amazon Web Services Regions where they log events. If you create a trail that logs events in all Amazon Web Services Regions, it will appear in the console in all Amazon Web Services Regions. If you create a trail that only logs events in a single Amazon Web Services Region, you can view and manage it only in that Amazon Web Services Region.
- To edit a trail in the list, choose the trail name.
- Configure at least one trail that applies to all Regions so that you receive log files from all Regions in your account.
- To log events from a specific Region and deliver log files to an S3 bucket in the same Region, you can update the trail to apply to a single Region. This is useful if you want to keep your log files separate. For example, you may want users to manage their own logs in specific Regions, or you may want to separate CloudWatch Logs alarms by Region.
- To log events from multiple Amazon accounts in one trail, consider creating an organization in Amazon Organizations and then creating an organization trail.
- Creating multiple trails will incur additional costs. For more information about prices, see Amazon CloudTrail Pricing.

Topics

- Managing CloudTrail costs (p. 149)
- CloudTrail naming requirements (p. 151)
- Amazon S3 bucket naming requirements (p. 151)
- Amazon KMS alias naming requirements (p. 151)

Managing CloudTrail costs

As a best practice, we recommend using Amazon services and tools that can help you manage CloudTrail costs. You can also configure and manage CloudTrail trails and event data stores in ways that capture the data you need while remaining cost-effective. For more information about CloudTrail pricing, see Amazon CloudTrail Pricing.

Tools to help manage costs

Amazon Budgets, a feature of Amazon Billing and Cost Management, lets you set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount.

As you create multiple trails or event data stores, creating a budget for CloudTrail by using Amazon Budgets is a recommended best practice, and can help you track your CloudTrail spending. Cost-based budgets help promote awareness of how much you might be billed for your CloudTrail use. Budget alerts notify you when your bill reaches a threshold that you define. When you receive a budget alert, you can make changes before the end of the billing cycle to manage your costs.

After you create a budget, you can use Amazon Cost Explorer to see how your CloudTrail costs are influencing your overall Amazon bill. In Amazon Cost Explorer, after adding CloudTrail to the **Service** filter, you can compare your historical CloudTrail spending to that of your current month-to-date (MTD)

spending, by both Region and account. This feature helps you monitor and detect unexpected costs in your monthly CloudTrail spending. Additional features in Cost Explorer let you compare CloudTrail spending to monthly spending at the specific resource level, providing information about what might be driving cost increases or decreases in your bill.

> **Note**
> Though you can apply tags to CloudTrail trails, Amazon Billing cannot currently use tags applied to trails for cost allocation. Cost Explorer can show costs for CloudTrail Lake event data stores and for the CloudTrail service as a whole.

To get started with Amazon Budgets, open Amazon Billing and Cost Management, and then choose **Budgets** in the left navigation bar. We recommend configuring budget alerts as you create a budget to track CloudTrail spending. For more information about how to use Amazon Budgets, see Managing Your Costs with Budgets and Best Practices for Amazon Budgets.

Trail configuration

CloudTrail offers flexibility in how you configure trails in your account. Some decisions that you make during the setup process require that you understand the impacts to your CloudTrail bill. The following are examples of how trail configurations can influence your CloudTrail bill.

Multiple trail creation

The first delivery of each management event for an account is free. If you create more trails that deliver the same management events to other destinations, those subsequent deliveries incur CloudTrail costs. You can do this to allow different user groups (such as developers, security personnel, and IT auditors) to receive their own copies of log files. For data events, all deliveries incur CloudTrail costs, including the first.

As you create more trails, it is especially important to be familiar with your logs, and understand the types and volumes of events that are generated by resources in your account. This helps you anticipate the volume of events that are associated with an account, and plan for trail costs. For example, using Amazon KMS-managed server-side encryption (SSE-KMS) on your S3 buckets can result in a large number of Amazon KMS management events in CloudTrail. Larger volumes of events across multiple trails can also influence costs.

To help limit the number of events that are logged to your trail, you can filter out Amazon KMS events by choosing **Exclude Amazon KMS events** on the **Create trail** or **Update trail** pages. The option to filter out events is only available if your trail is logging management events. For more information, see Creating a trail (p. 78) or Updating a trail (p. 89) in this guide.

Amazon Organizations

When you set up an Organizations trail with CloudTrail, CloudTrail replicates the trail to each member account within your organization. The new trail is created *in addition to* any existing trails in member accounts. Be sure that the configuration of your organization trail matches how you want trails configured for all accounts within an organization, because the organization trail configuration propagates to all accounts.

Because Organizations creates a trail in each member account, an individual member account that creates an additional trail to collect the same management events as the Organizations trail is collecting a second copy of events. The account is charged for the second copy. Similarly, if an account has a multi-Region trail, and creates a second trail in a single Region to collect the same management events as the multi-Region trail, the trail in the single Region is delivering a second copy of events. The second copy incurs charges.

See also

- Amazon CloudTrail Pricing

- Managing Your Costs with Budgets
- Getting Started with Cost Explorer
- Prepare for creating a trail for your organization (p. 127)

CloudTrail naming requirements

CloudTrail resource names must meet the following requirements:

- Contain only ASCII letters (a-z, A-Z), numbers (0-9), periods (.), underscores (_), or dashes (-).
- Start with a letter or number, and end with a letter or number.
- Be between 3 and 128 characters.
- Have no adjacent periods, underscores or dashes. Names like my-_namespace and my-\-namespace are invalid.
- Not be in IP address format (for example, `192.168.5.4`).

Amazon S3 bucket naming requirements

The Amazon S3 bucket that you use to store CloudTrail log files must have a name that conforms with naming requirements for non-US Standard Regions. Amazon S3 defines a bucket name as a series of one or more labels, separated by periods, that adhere to the following rules:

For complete information about Amazon S3 bucket naming requirements, see Bucket naming rules in the *Amazon Simple Storage Service User Guide*

The following are some of the bucket naming rules:

- The bucket name can be between 3 and 63 characters long, and can contain only lower-case characters, numbers, periods, and dashes.
- Each label in the bucket name must start with a lowercase letter or number.
- The bucket name cannot contain underscores, end with a dash, have consecutive periods, or use dashes adjacent to periods.
- The bucket name cannot be formatted as an IP address (198.51.100.24).

> **Warning**
> Because S3 allows your bucket to be used as a URL that can be accessed publicly, the bucket name that you choose must be globally unique. If some other account has already created a bucket with the name that you chose, you must use another name. For more information, see Bucket restrictions and limitations in the *Amazon Simple Storage Service User Guide*.

Amazon KMS alias naming requirements

When you create an Amazon KMS key, you can choose an alias to identify it. For example, you might choose the alias "KMS-CloudTrail-us-west-2" to encrypt the logs for a specific trail.

The alias must meet the following requirements:

- Between 1 and 256 characters, inclusive
- Contain alphanumeric characters (A-Z, a-z, 0-9), hyphens (-), forward slashes (/), and underscores (_)
- Cannot begin with **aws**

For more information, see Creating Keys in the *Amazon Key Management Service Developer Guide*.

Using Amazon CloudTrail with interface VPC endpoints

If you use Amazon Virtual Private Cloud (Amazon VPC) to host your Amazon resources, you can establish a private connection between your VPC and Amazon CloudTrail. You can use this connection to enable CloudTrail to communicate with your resources on your VPC without going through the public internet.

Amazon VPC is an Amazon service that you can use to launch Amazon resources in a virtual network that you define. With a VPC, you have control over your network settings, such the IP address range, subnets, route tables, and network gateways. With VPC endpoints, the routing between the VPC and Amazon services is handled by the Amazon network, and you can use IAM policies to control access to service resources.

To connect your VPC to CloudTrail, you define an *interface VPC endpoint* for CloudTrail. An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported Amazon service. The endpoint provides reliable, scalable connectivity to CloudTrail without requiring an internet gateway, network address translation (NAT) instance, or VPN connection. For more information, see What is Amazon VPC in the *Amazon VPC User Guide*.

Interface VPC endpoints are powered by Amazon PrivateLink, an Amazon technology that enables private communication between Amazon services using an elastic network interface with private IP addresses. For more information, see Amazon PrivateLink.

The following steps are for users of Amazon VPC. For more information, see Get started with Amazon VPC in the *Amazon VPC User Guide*.

Availability

CloudTrail currently supports VPC endpoints in the following Amazon Regions:

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Melbourne)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Milan)

- Europe (Paris)
- Europe (Spain)
- Europe (Stockholm)
- Europe (Zurich)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- Middle East (UAE)
- South America (São Paulo)
- Amazon GovCloud (US-East)
- Amazon GovCloud (US-West)

Create a VPC endpoint for CloudTrail

To start using CloudTrail with your VPC, create an interface VPC endpoint for CloudTrail. For more information, see Access an Amazon Web Service using an interface VPC endpoint in the *Amazon VPC User Guide*.

You don't need to change the settings for CloudTrail. CloudTrail calls other Amazon Web Services using either public endpoints or private interface VPC endpoints, whichever are in use.

Shared subnets

A CloudTrail VPC endpoint, like any other VPC endpoint, can only be created by an owner account in the shared subnet. However, a participant account can use CloudTrail VPC endpoints in subnets that are shared with the participant account. For more information about Amazon VPC sharing, see Share your VPC with other accounts in the *Amazon VPC User Guide*.

Viewing service-linked channels for CloudTrail by using the Amazon CLI

Amazon services can create a service-linked channel to receive CloudTrail events on your behalf. The Amazon service creating the service-linked channel configures advanced event selectors for the channel and specifies whether the channel applies to all Regions, or the current Region.

Using the Amazon CLI, you can view information about any CloudTrail service-linked channels created by Amazon services.

Topics

- Get a CloudTrail service-linked channel (p. 153)
- List all CloudTrail service-linked channels (p. 154)
- Amazon service events on service-linked channels (p. 155)

Get a CloudTrail service-linked channel

The following example Amazon CLI command returns information about a specific CloudTrail service-linked channel, including the name of the destination Amazon service, any advanced selectors configured for the channel, and whether the channel applies to all Regions or a single Region.

You must specify an ARN or the ID suffix of an ARN for `--channel`.

```
aws cloudtrail get-channel --channel EXAMPLE-ee54-4813-92d5-999aeEXAMPLE
```

The following is an example response. In this example, `Amazon_service_name` represents the name of the Amazon service that created the channel.

```
{
    "ChannelArn": "arn:aws:cloudtrail:us-east-1:111122223333:channel/EXAMPLE-
ee54-4813-92d5-999aeEXAMPLE",
    "Name": "aws-service-channel/Amazon_service_name/slc",
    "Source": "CloudTrail",
    "SourceConfig": {
        "ApplyToAllRegions": false,
        "AdvancedEventSelectors": [
            {
                "Name": "Management Events Only",
                "FieldSelectors": [
                    {
                        "Field": "eventCategory",
                        "Equals": [
                            "Management"
                        ]
                    }
                ]
            }
        ]
    },
    "Destinations": [
        {
            "Type": "Amazon_SERVICE",
            "Location": "Amazon_service_name"
        }
    ]
}
```

List all CloudTrail service-linked channels

The following example Amazon CLI command returns information about all CloudTrail service-linked channels that were created on your behalf. Optional parameters include `--max-results`, to specify a maximum number of results that you want the command to return on a single page. If there are more results than your specified `--max-results` value, run the command again adding the returned `NextToken` value to get the next page of results.

```
aws cloudtrail list-channels
```

The following is an example response. In this example, `Amazon_service_name` represents the name of the Amazon service that created the channel.

```
{
    "Channels": [
        {
            "ChannelArn": "arn:aws:cloudtrail:us-east-1:111122223333:channel/EXAMPLE-
ee54-4813-92d5-999aeEXAMPLE",
            "Name": "aws-service-channel/Amazon_service_name/slc"
        }
```

```
    ]
}
```

Amazon service events on service-linked channels

The Amazon service managing the service-linked channel can initiate actions on the service-linked channel (for example, creating or updating a service-linked channel). CloudTrail logs these actions as Amazon service events, and delivers these events to the **Event history**, and any active trails and event data stores configured for management events. For these events, the `eventType` field is `AwsServiceEvent`.

The following is an example log file entry of an Amazon service event for creation of a service-linked channel.

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "accountId":"111122223333",
      "invokedBy":"AWS Internal"
   },
   "eventTime":"2022-08-18T17:11:22Z",
   "eventSource":"cloudtrail.amazonaws.com",
   "eventName":"CreateServiceLinkedChannel",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"AWS Internal",
   "requestParameters":null,
   "responseElements":null,
   "requestID":"564f004c-EXAMPLE",
   "eventID":"234f004b-EXAMPLE",
   "readOnly":false,
   "resources":[
      {
         "accountId":"184434908391",
         "type":"AWS::CloudTrail::Channel",
         "ARN":"arn:aws:cloudtrail:us-east-1:111122223333:channel/7944f0ec-EXAMPLE"
      }
   ],
   "eventType":"AwsServiceEvent",
   "managementEvent":true,
   "recipientAccountId":"111122223333",
   "eventCategory":"Management"
}
```

Working with Amazon CloudTrail Lake

Amazon CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into event data stores, which are immutable collections of events based on criteria that you select by applying advanced event selectors (p. 255). You can keep the event data in an event data store for up to seven years, or 2557 days. By default, event data is retained for the maximum period, 2557 days. The selectors that you apply to an event data store control which events persist and are available for you to query. CloudTrail Lake is an auditing solution that can complement your compliance stack, and assist you with near real-time troubleshooting.

You can use CloudTrail Lake *integrations* to log and store user activity data from outside of Amazon; from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers. After you create event data stores in CloudTrail Lake and create a channel to log activity events, you call the `PutAuditEvents` API to ingest your application activity into CloudTrail. You can then use CloudTrail Lake to search, query, and analyze the data that is logged from your applications.

Integrations can also log events to your event data stores from over a dozen CloudTrail partners. In a partner integration, you create destination event data stores, a channel, and a resource policy. After you create the integration, you provide the channel ARN to the partner. There are two types of integrations: direct and solution. With direct integrations, the partner calls the `PutAuditEvents` API to deliver events to the event data store for your Amazon account. With solution integrations, the application runs in your Amazon account and the application calls the `PutAuditEvents` API to deliver events to the event data store for your Amazon account.

When you create an event data store, you choose the type of events to include in your event data store. You can create an event data store to include CloudTrail events, Amazon Config configuration items, Amazon Audit Manager evidence, or events from outside of Amazon. Each event data store can only contain a specific event category (for example, Amazon Config configuration items), because the event schema is unique to the event category. You can run SQL queries across multiple event data stores using the supported SQL JOIN keywords. For information about running queries across multiple event data stores, see Advanced, multi-table query support (p. 229).

CloudTrail Lake queries offer a deeper and more customizable view of events than simple key and value lookups in **Event history**, or running `LookupEvents`. An **Event history** search is limited to a single Amazon Web Services account, only returns events from a single Amazon Web Services Region, and cannot query multiple attributes. In contrast, CloudTrail Lake users can run complex Standard Query Language (SQL) queries across multiple event fields. CloudTrail Lake supports all valid Presto `SELECT` statements and functions. For more information about the supported SQL functions and operators, see Functions and Operators on the Presto 0.280 documentation website.

You can save CloudTrail Lake queries for future use, and view results of queries for up to seven days. When you run queries, you can save the query results to an Amazon S3 bucket. CloudTrail Lake can also store events from an organization in Amazon Organizations in an event data store, including events from multiple Regions and accounts.

You can use CloudTrail Lake dashboards to visualize the data in your event data stores. Each dashboard consists of multiple widgets and each widget represents a SQL query. For more information about Lake dashboards, see View Lake dashboards (p. 191).

CloudTrail doesn't support authorization based on tags for trails. However, you can control access to actions on event data stores by using authorization based on tags. For more information and examples, see Examples: Denying access to create or delete event data stores based on tags (p. 354) in this guide.

By default, all events in an event data store are encrypted by CloudTrail. When you configure an event data store, you can choose to use your own Amazon Key Management Service key. Using your own KMS key incurs Amazon KMS costs for encryption and decryption. After you associate an event data store with a KMS key, the KMS key cannot be removed or changed.

CloudTrail Lake event data stores incur charges when they ingest events and when you run queries. For event data stores containing CloudTrail events (data and management events), or Amazon Config configuration items, you can choose whether you want the event data store to start ingesting events when you create it, and you can also stop ingestion at any time. For more information about CloudTrail Lake pricing, see Amazon CloudTrail Pricing.

CloudTrail Lake supports Amazon CloudWatch metrics, which you can use to view information about the amount of data ingested into your event data store during the last hour and over the course of its retention period. For more information about supported CloudWatch metrics, see Supported CloudWatch metrics (p. 235).

> **Note**
> CloudTrail typically delivers events within an average of about 5 minutes of an API call. This time is not guaranteed.

Topics

- CloudTrail Lake supported Regions (p. 157)
- Create an event data store (p. 159)
- Manage event data store lifecycles (p. 172)
- Copy trail events to an event data store (p. 173)
- Create an integration with an event source outside of Amazon (p. 180)
- View Lake dashboards (p. 191)
- Create or edit a query (p. 193)
- Learning resources (p. 213)
- Managing CloudTrail Lake by using the Amazon CLI (p. 214)
- CloudTrail Lake SQL constraints (p. 227)
- Supported CloudWatch metrics (p. 235)

CloudTrail Lake supported Regions

Currently, CloudTrail Lake is supported in the following Amazon Web Services Regions:

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cloudtrail.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cloudtrail.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	cloudtrail.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cloudtrail.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	cloudtrail.ca-central-1.amazonaws.com	HTTPS
Africa (Cape Town)	af-south-1	cloudtrail.af-south-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Hong Kong)	ap-east-1	cloudtrail.ap-east-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	cloudtrail.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cloudtrail.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cloudtrail.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Osaka)	ap-northeast-3	cloudtrail.ap-northeast-3.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	cloudtrail.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cloudtrail.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Jakarta)	ap-southeast-3	cloudtrail.ap-southeast-3.amazonaws.com	HTTPS
Europe (Frankfurt)	eu-central-1	cloudtrail.eu-central-1.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	cloudtrail.eu-north-1.amazonaws.com	HTTPS
Europe (Ireland)	eu-west-1	cloudtrail.eu-west-1.amazonaws.com	HTTPS
Europe (London)	eu-west-2	cloudtrail.eu-west-2.amazonaws.com	HTTPS
Europe (Paris)	eu-west-3	cloudtrail.eu-west-3.amazonaws.com	HTTPS
Europe (Milan)	eu-south-1	cloudtrail.eu-south-1.amazonaws.com	HTTPS
Middle East (UAE)	me-central-1	cloudtrail.me-central-1.amazonaws.com	HTTPS
Middle East (Bahrain)	me-south-1	cloudtrail.me-south-1.amazonaws.com	HTTPS
Amazon GovCloud (US-East)	us-gov-east-1	cloudtrail.us-gov-east-1.amazonaws.com	HTTPS
Amazon GovCloud (US-West)	us-gov-west-1	cloudtrail.us-gov-west-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	cloudtrail.sa-east-1.amazonaws.com	HTTPS

For more information about using CloudTrail in the Amazon GovCloud (US-East) Region, see Amazon GovCloud (US-East) Endpoints in the *Amazon GovCloud (US) User Guide*.

Create an event data store

When you create an event data store in CloudTrail Lake, you choose the type of events to include in your event data store. You can create an event data store to include CloudTrail data or management events, Amazon Config configuration items, or events outside of Amazon. Each event data store type can only contain specific event categories (for example, Amazon Config configuration items), because the event schema is unique to the event category. You can run SQL queries across multiple event data stores using the supported SQL JOIN keywords. For information about running queries across multiple event data stores, see Advanced, multi-table query support (p. 229).

The following table shows the supported event categories for each event data store type. The **eventCategory** column shows the value that you would specify in the advanced event selectors to collect events of that type.

Event type (console)	eventCategory (API)	Description

CloudTrail events	`Management`
	
	`Data`
		This event data store type can collect CloudTrail management and data events. For more information, see Create an event data store for CloudTrail events.

Configuration items	`ConfigurationItem`	This event data store type can collect Amazon Config configuration items. For more information, see Create an event data store for Amazon Config configuration items.
Events from integration	`ActivityAuditLog`	This event data store type can collect non-Amazon events from integrations. For more information, see Create an event data store for events outside of Amazon.

You can also create an event data store for Amazon Audit Manager evidence by using the Audit Manager console. For more information about aggregating evidence in CloudTrail Lake using Audit Manager, see Understanding how evidence finder works with CloudTrail Lake in the *Amazon Audit Manager User Guide*.

The sections which follow describe how to create an event data store using the CloudTrail console. For information about how to create an event data store using the Amazon CLI, see Managing CloudTrail Lake by using the Amazon CLI (p. 214).

Topics

- Create an event data store for CloudTrail events (p. 159)
- Create an event data store for Amazon Config configuration items (p. 163)
- Create an event data store for events outside of Amazon (p. 170)

Create an event data store for CloudTrail events

Event data stores for CloudTrail events can log CloudTrail management and data events. You can keep the event data in an event data store for up to seven years, or 2557 days. By default, event data is retained for 2557 days, and termination protection is enabled for an event data store.

An event data store incurs charges when it ingest events and when you run queries. You can stop an event data store from ingesting events at any time and you can restart ingestion at any time. When you

create an event data store, you can choose whether you want the event data store to immediately start ingesting events. For more information about CloudTrail Lake pricing, see Amazon CloudTrail Pricing.

To create an event data store for CloudTrail events

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Event data stores**.
3. Choose **Create event data store**.
4. On the **Configure event data store** page, in **General details**, enter a name for the event data store. A name is required.
5. Specify a retention period for the event data store. You can choose a value of 7 years, 2 years, 1 year, or 6 months, or define a custom value. By default, event data is retained for 2557 days, which is the maximum value.

   CloudTrail Lake determines whether to retain an event by checking if the `eventTime` of the event is within the specified retention period. For example, if you specify a retention period of 90 days, CloudTrail will remove events when their `eventTime` is older than 90 days.

   Note
   If you are copying trail events to this event data store, CloudTrail will not copy an event if its `eventTime` is older than the specified retention period.
   We recommend that when you choose a retention period, you consider both the age of the events that you want to copy as well as how long you want to keep the copied events in your event data store. For example, if you copy trail events that are 5 years old and specify a retention period of 7 years, the event data store will retain those events for two years.
6. (Optional) To enable encryption using Amazon Key Management Service, choose **Use my own Amazon KMS key**. Choose **New** to have an Amazon KMS key created for you, or choose **Existing** to use an existing KMS key. In **Enter KMS alias**, specify an alias, in the format `alias/MyAliasName`. Using your own KMS key requires that you edit your KMS key policy to allow CloudTrail logs to be encrypted and decrypted. For more information, see Configure Amazon KMS key policies for CloudTrail (p. 387). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Using your own KMS key incurs Amazon KMS costs for encryption and decryption. After you associate an event data store with a KMS key, the KMS key cannot be removed or changed.

   Note
   To enable Amazon Key Management Service encryption for an organization event data store, you must use an existing KMS key for the management account.
7. (Optional) In the **Tags** section, you can add up to 50 tag key pairs to help you identify, sort, and control access to your event data store. For more information about how to use IAM policies to authorize access to an event data store based on tags, see Examples: Denying access to create or delete event data stores based on tags (p. 354). For more information about how you can use tags in Amazon, see Tagging Amazon resources in the *Amazon Web Services General Reference*.
8. Choose **Next** to configure the event data store.
9. On the **Choose events** page, choose **Amazon events**, and then choose **CloudTrail events**.
10. For **CloudTrail events**, choose at least one event type. By default, **Management events** is selected. You can add both management and data events to your event data store. For more information about management events, see Logging management events (p. 238). For more information about data events, see Logging data events (p. 242).
11. (Optional) Choose **Copy trail events** if you want to copy events from an existing trail to run queries on past events. To copy trail events to an organization event data store, you must use the management account for the organization. The delegated administrator account cannot copy trail

events to an organization event data store. For more information about considerations for copying trail events, see Considerations for copying trail events (p. 174).

12. To have your event data store collect events from all accounts in an Amazon Organizations organization, select **Enable for all accounts in my organization**. You must be signed in to the management account or delegated administrator account for the organization to create an event data store that collects events for an organization.
13. Expand **Additional settings** to choose whether you want your event data store to collect events for all Amazon Web Services Regions, or only the current Amazon Web Services Region, and choose whether the event data store ingests events. By default, your event data store collects events from all Regions in your account and starts ingesting events when it's created.

    a. Select **Include only the current region in my event data store** to include only events that are logged in the current Region. If you do not choose this option, your event data store includes events from all Regions.

    b. Deselect **Ingest events** if you do not want the event data store to start ingesting events. For example, you may want to deselect **Ingest events**, if you are copying trail events and do not want the event data store to include any future events. By default, the event data store starts ingesting events when it's created.
14. If your event data store includes management events, you can choose from the following options. For more information about management events, see Logging management events (p. 238).

    a. Choose whether you want to include **Read** events, **Write** events, or both. At least one is required.

    b. Choose whether to exclude Amazon Key Management Service or Amazon RDS Data API events from your event data store.
15. To include data events in your event data store, do the following.

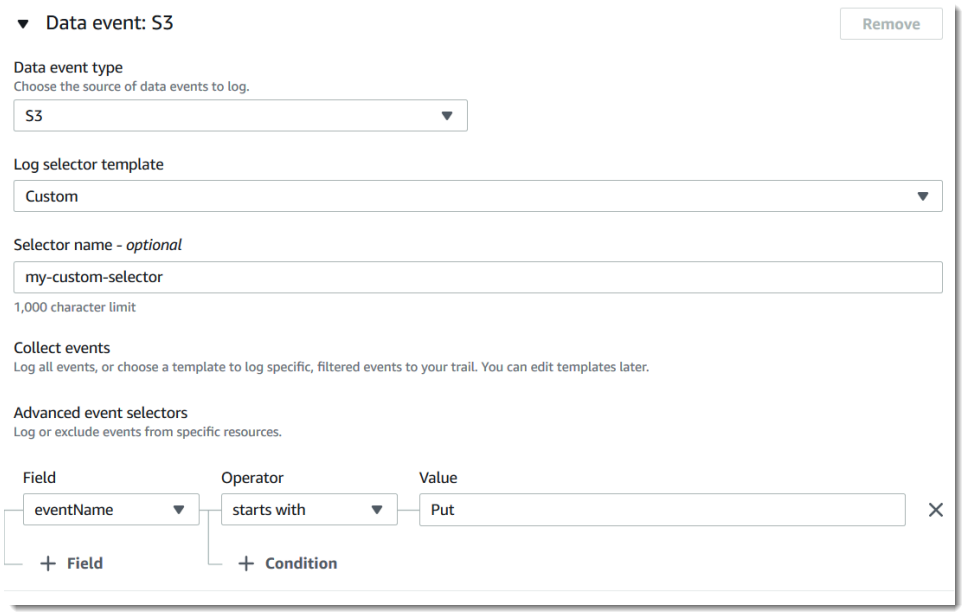
    a. Choose a data event type. This is the Amazon Web Service and resource on which data events are logged. To log data events for Amazon Glue tables created by Lake Formation, choose **Lake Formation** for the data type.

    b. In **Log selector template**, choose a template. You can choose to log all data events, `readOnly` events, `writeOnly` events, or **Custom** to build a custom log selector.

    c. (Optional) In **Selector name**, enter a name to identify your selector. The selector name is a descriptive name for an advanced event selector, such as "Log data events for only two S3 buckets". The selector name is listed as `Name` in the advanced event selector and is viewable if you expand the **JSON view**.

    d. In **Advanced event selectors**, build expressions by choosing values for **Field**, **Operator**, and **Value**. Advanced event selectors for an event data store work the same as advanced event selectors that you apply to a trail. For more information about how to build advanced event selectors, see Logging data events with advanced event selectors (p. 255).

       The following example uses a **Custom** log selector template to choose only event names from S3 objects that start with `Put`, such as `PutObject`. Because the advanced event selector does not include or exclude any other event types or resource ARNs, all S3 data events, both read and write, that are logged to the US East (N. Virginia) Region, and that have event names starting with `Put`, are stored in the event data store.



Important
To exclude or include data events with advanced event selectors by using an S3 bucket ARN, always use the **Starts with** operator.

e. Optionally, expand **JSON view** to see your advanced event selectors as a JSON block.

16. To copy existing trail events to your event data store, do the following.

    a. Choose the trail that you want to copy. By default, CloudTrail only copies CloudTrail events contained in the S3 bucket's `CloudTrail` prefix and the prefixes inside the `CloudTrail` prefix, and does not check prefixes for other Amazon services. If you want to copy CloudTrail events contained in another prefix, choose **Enter S3 URI**, and then choose **Browse S3** to browse to the prefix. If the source S3 bucket for the trail uses a KMS key for data encryption, ensure that the KMS key policy allows CloudTrail to decrypt the data. If your source S3 bucket uses multiple KMS keys, you must update each key's policy to allow CloudTrail to decrypt the data in the bucket. For more information about updating the KMS key policy, see KMS key policy for decrypting data in the source S3 bucket (p. 177).

    b. (Optional) Choose a time range for copying the events. If you choose a time range, CloudTrail checks the prefix and log file name to verify the name contains a date between the chosen start and end date before attempting to copy trail events. You can choose a **Relative range** or an **Absolute range**. To avoid duplicating events between the source trail and destination event data store, choose a time range that is earlier than the creation of the event data store.

    - If you choose **Relative range**, you can choose to copy events logged in the last 6 months, 1 year, 2 years, 7 years, or a custom range. CloudTrail copies the events logged within the chosen time period.
    - If you choose **Absolute range**, you can choose a specific start and end date. CloudTrail copies the events that occurred between the chosen start and end dates.

    c. For **Permissions**, choose from the following IAM role options. If you choose an existing IAM role, verify that the IAM role policy provides the necessary permissions. For more information about updating the IAM role permissions, see IAM permissions for copying trail events (p. 175).

    - Choose **Create a new role (recommended)** to create a new IAM role. For **Enter IAM role name**, enter a name for the role. CloudTrail automatically creates the necessary permissions for this new role.

- Choose **Use a custom IAM role** to use a custom IAM role that is not listed. For **Enter IAM role ARN**, enter the IAM ARN.
- Choose **Use an existing role** to choose an existing IAM role from the drop-down list.

17. Choose **Next** to review your choices.
18. On the **Review and create** page, review your choices. Choose **Edit** to make changes to a section. When you're ready to create the event data store, choose **Create event data store**.
19. The new event data store is visible in the **Event data stores** table on the **Event data stores** page.

    From this point forward, the event data store captures events that match its advanced event selectors (if you selected the **Ingest events** option). Events that occurred before you created the event data store are not in the event data store, unless you opted to copy existing trail events.

You can now run queries on your new event data store. The **Sample queries** tab provides example queries to get you started. For more information about creating and editing queries, see Create or edit a query (p. 193).

You can also view the CloudTrail Lake dashboard to visualize the events in your event data store. For more information about Lake dashboards, see View Lake dashboards (p. 191).

Create an event data store for Amazon Config configuration items

You can create an event data store to include Amazon Config configuration items, and use the event data store to investigate non-compliant changes to your production environments. With an event data store, you can relate non-compliant rules to the users and resources associated with the changes. A configuration item represents a point-in-time view of the attributes of a supported Amazon resource that exists in your account. Amazon Config creates a configuration item whenever it detects a change to a resource type that it is recording. Amazon Config also creates configuration items when a configuration snapshot is captured.

You can use both Amazon Config and CloudTrail Lake to run queries against your configuration items. You can use Amazon Config to query the current configuration state of Amazon resources based on configuration properties for a single Amazon Web Services account and Amazon Web Services Region, or across multiple accounts and Regions. In contrast, you can use CloudTrail Lake to query across diverse data sources such as CloudTrail events, configuration items, and rule evaluations. CloudTrail Lake queries cover all Amazon Config configuration items including resource configuration and compliance history.

Creating an event data store for configuration items doesn't impact existing Amazon Config advanced queries, or any configured Amazon Config aggregators. You can continue to run advanced queries using Amazon Config, and Amazon Config continues to deliver history files to your S3 buckets.

Limitations

The following limitations apply to event data stores for configuration items.

- No support for custom configuration items
- No support for event filtering using advanced event selectors

Prerequisites

Before you create your event data store, set up Amazon Config recording for all your accounts and Regions. You can use Quick Setup, a capability of Amazon Systems Manager, to quickly create a configuration recorder powered by Amazon Config.

Note
You are charged service usage fees when Amazon Config starts recording configurations. For more information about pricing, see Amazon Config Pricing. For information about managing the configuration recorder, see Managing the Configuration Recorder in the *Amazon Config Developer Guide*.

Additionally, the following actions are recommended, but are not required to create an event data store.

- Set up an Amazon S3 bucket to receive a configuration snapshot on request and configuration history. For more information about snapshots, see Managing the Delivery Channel and Delivering Configuration Snapshot to an Amazon S3 Bucket in the *Amazon Config Developer Guide*.
- Specify the rules that you want Amazon Config to use to evaluate compliance information for the recorded resource types. Several of the CloudTrail Lake sample queries for Amazon Config require Amazon Config Rules to evaluate the compliance state of your Amazon resources. For more information about Amazon Config Rules, see Evaluating Resources with Amazon Config Rules in the *Amazon Config Developer Guide*.

To create an event data store for configuration items

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Event data stores**.
3. Choose **Create event data store**.
4. On the **Configure event data store** page, in **General details**, enter a name for the event data store. A name is required.
5. Specify a retention period for the event data store in days. Valid values are integers from 7 to 2557 (seven years). The event data store retains configuration items for the specified number of days. By default, event data is retained for 2557 days.
6. (Optional) To enable encryption using Amazon Key Management Service, choose **Use my own Amazon KMS key**. Choose **New** to have an Amazon KMS key created for you, or choose **Existing** to use an existing KMS key. In **Enter KMS alias**, specify an alias, in the format `alias/MyAliasName`. Using your own KMS key requires that you edit your KMS key policy to allow CloudTrail logs to be encrypted and decrypted. For more information, see Configure Amazon KMS key policies for CloudTrail (p. 387). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Using your own KMS key incurs Amazon KMS costs for encryption and decryption. After you associate an event data store with a KMS key, the KMS key cannot be removed or changed.

   Note
   To enable Amazon Key Management Service encryption for an organization event data store, you must use an existing KMS key for the management account.
7. (Optional) In the **Tags** section, you can add up to 50 tag key pairs to help you identify, sort, and control access to your event data store. For more information about how to use IAM policies to authorize access to an event data store based on tags, see Examples: Denying access to create or delete event data stores based on tags (p. 354). For more information about how you can use tags in Amazon, see Tagging Amazon resources in the *Amazon Web Services General Reference*.
8. Choose **Next**.
9. On the **Choose events** page, choose **Amazon events**, and then choose **Configuration items**.
10. CloudTrail stores the event data store resource in the Region in which you create it, but by default, the configuration items collected in the data store are from all Regions in your account that have recording enabled. Optionally, you can select **Include only the current region in my event data store** to include only configuration items that are captured in the current Region. If you do not

choose this option, your event data store includes configuration items from all Regions that have recording enabled.

11. To have your event data store collect configuration items from all accounts in an Amazon Organizations organization, select **Enable for all accounts in my organization**. You must be signed in to the management account or delegated administrator account for the organization to create an event data store that collects configuration items for an organization.
12. Choose **Next** to review your choices.
13. On the **Review and create** page, review your choices. Choose **Edit** to make changes to a section. When you're ready to create the event data store, choose **Create event data store**.
14. The new event data store is visible in the **Event data stores** table on the **Event data stores** page.

    From this point forward, the event data store captures configuration items. Configuration items that occurred before you created the event data store are not in the event data store.

Sample queries

You can now run queries on your new event data store. The **Sample queries** tab on the CloudTrail console provides example queries to get you started. The following are a few of the sample queries that you can run against your configuration item event data store.

Description	Query
Find which user performed an action that resulted in a non-compliant status by joining a configuration item event data store with a CloudTrail event data store.	SELECT element_at(config1.eventData.configuration, 'targetResourceId') as targetResourceId, element_at(config1.eventData.configuration, 'complianceType') as complianceType, config2.eventData.resourceType, cloudtrail.userIdentity FROM *config_event_data_store_ID* as config1 JOIN *config_event_data_store_ID* as config2 on element_at(config1.eventData.configuration, 'targetResourceId') = config2.eventData.resourceId JOIN *cloudtrail_event_data_store_ID* as cloudtrail on config2.eventData.arn = element_at(cloudtrail.resources, 1).arn WHERE element_at(config1.eventData.configuration, 'configRuleList') is not null AND element_at(config1.eventData.configuration, 'complianceType') = 'NON_COMPLIANT' AND cloudtrail.eventTime > '2022-11-14 00:00:00' AND config2.eventData.resourceType = 'AWS::DynamoDB::Table'

Description	Query
Find all Amazon Config rules and return the compliance state from configuration items generated within the past day.	`SELECT eventData.configuration, eventData.accountId, eventData.awsRegion, eventData.resourceName, eventData.resourceCreationTime, element_at(eventData.configuration,'complianceType') AS complianceType, element_at(eventData.configuration, 'configRuleList') AS configRuleList, element_at(eventData.configuration, 'resourceId') AS resourceId, element_at(eventData.configuration, 'resourceType') AS resourceType FROM config_event_data_store_ID WHERE eventData.resourceType = 'AWS::Config::ResourceCompliance' AND eventTime > '2022-11-22 00:00:00' ORDER BY eventData.resourceCreationTime DESC limit 10`
Find the total count of Amazon Config resources grouped by resource type, account ID, and Region.	`SELECT eventData.resourceType, eventData.awsRegion, eventData.accountId, COUNT (*) AS resourceCount FROM config_event_data_store_ID WHERE eventTime > '2022-11-22 00:00:00' GROUP BY eventData.resourceType, eventData.awsRegion, eventData.accountId`
Find the resource creation time for all Amazon Config configuration items generated on a specific date.	`SELECT eventData.configuration, eventData.accountId, eventData.awsRegion, eventData.resourceId, eventData.resourceName, eventData.resourceType, eventData.availabilityZone, eventData.resourceCreationTime FROM config_event_data_store_ID WHERE eventTime > '2022-11-16 00:00:00' AND eventTime < '2022-11-17 00:00:00' ORDER BY eventData.resourceCreationTime DESC limit 10;`

For more information about creating and editing queries, see Create or edit a query (p. 193).

Configuration item schema

The following table describes the required and optional schema elements that match those in configuration item records. The contents of `eventData` are provided by your configuration items; other fields are provided by CloudTrail after ingestion.

CloudTrail event record contents are described in more detail in CloudTrail record contents (p. 402).

- Fields that are provided by CloudTrail after ingestion (p. 167)
- Fields that are provided by your events (p. 168)

Fields that are provided by CloudTrail after ingestion

Field name	Input type	Requirement	Description
eventVersion	string	Required	The version of the Amazon event format.
eventCategory	string	Required	The event category. For configuration items, the valid value is `ConfigurationItem`.
eventType	string	Required	The event type. For configuration items, the valid value is `AwsConfigurationItem`.
eventID	string	Required	A unique ID for an event.
eventTime	string	Required	The event timestamp, in `yyyy-MM-DDTHH:mm:ss` format, in Universal Coordinated Time (UTC).
awsRegion	string	Required	The Amazon Web Services Region to which to assign an event.
recipientAccountId	string	Required	Represents the Amazon Web Services account ID that received this event.
addendum	addendum	Optional	Shows information about why an event was delayed. If information was missing from an existing event, the addendum block includes the missing

Field name	Input type	Requirement	Description
			information and a reason for why it was missing.

Fields in `eventData` are provided by your configuration items

Field name	Input type	Requirement	Description
eventData	-	Required	Fields in eventData are provided by your configuration items.
• configurationItemVersion	string	Optional	The version of the configuration item from its source.
• configurationItemCaptureTime	string	Optional	The time when the configuration recording was initiated.
• configurationItemStatus	string	Optional	The configuration item status. Valid values are OK, `ResourceDiscovered`, `ResourceNotRecorded`, `ResourceDeleted`, and `ResourceDeletedNotRecorded`.
• accountId	string	Optional	The 12-digit Amazon Web Services account ID associated with the resource.
• resourceType	string	Optional	The type of Amazon resource. For more information about valid resource types, see ConfigurationItem in the *Amazon Config API Reference*.
• resourceId	string	Optional	The ID of the resource (for example., sg-*xxxxxx*).
• resourceName	string	Optional	The custom name of the resource, if available.
• arn	string	Optional	Amazon Resource Name (ARN) associated with the resource.

Field name	Input type	Requirement	Description
• awsRegion	string	Optional	The Amazon Web Services Region where the resource resides.
• availabilityZone	string	Optional	The Availability Zone associated with the resource.
• resourceCreationTime	string	Optional	The time stamp when the resource was created.
• configuration	JSON	Optional	The description of the resource configuration.
• supplementaryConfiguration	JSON	Optional	Configuration attributes that Amazon Config returns for certain resource types to supplement the information returned for the configuration parameter.
• relatedEvents	string	Optional	A list of CloudTrail event IDs.
• relationships	-	Optional	A list of related Amazon resources.
• • name	string	Optional	The type of relationship with the related resource.
• • resourceType	string	Optional	The resource type of the related resource.
• • resourceId	string	Optional	The ID of the related resource (for example, sg-*xxxxxx*).
• • resourceName	string	Optional	The custom name of the related resource, if available.
• tags	JSON	Optional	A mapping of key value tags associated with the resource.

The following example shows the hierarchy of schema elements that match those in configuration item records.

```
{
  "eventVersion": String,
  "eventCategory: String,
  "eventType": String,
  "eventID": String,
```

```
  "eventTime": String,
  "awsRegion": String,
  "recipientAccountId": String,
  "addendum": Addendum,
  "eventData": {
      "configurationItemVersion": String,
      "configurationItemCaptureTime": String,
      "configurationItemStatus": String,
      "configurationStateId": String,
      "accountId": String,
      "resourceType": String,
      "resourceId": String,
      "resourceName": String,
      "arn": String,
      "awsRegion": String,
      "availabilityZone": String,
      "resourceCreationTime": String,
      "configuration": {
        JSON,
      },
      "supplementaryConfiguration": {
        JSON,
      },
      "relatedEvents": [
        String
      ],
      "relationships": [
        struct{
          "name" : String,
          "resourceType": String,
          "resourceId": String,
          "resourceName": String
        }
      ],
     "tags": {
       JSON
     }
    }
  }
}
```

Create an event data store for events outside of Amazon

You can create an event data store to include events outside of Amazon, and then use CloudTrail Lake to search, query, and analyze the data that is logged from your applications.

You can use CloudTrail Lake *integrations* to log and store user activity data from outside of Amazon; from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers.

When you create an event data store for an integration, you also create a channel, and attach a resource policy to the channel.

To create an event data store for events outside of Amazon

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Event data stores**.
3. Choose **Create event data store**.

4. On the **Configure event data store** page, in **General details**, enter a name for the event data store. A name is required.
5. Specify a retention period for the event data store in days. Valid values are integers between 7 to 2557 (seven years). The event data store retains event data for the specified number of days. By default, event data is retained for 2557 days.
6. (Optional) To enable encryption using Amazon Key Management Service, choose **Use my own Amazon KMS key**. Choose **New** to have an Amazon KMS key created for you, or choose **Existing** to use an existing KMS key. In **Enter KMS alias**, specify an alias, in the format `alias/MyAliasName`. Using your own KMS key requires that you edit your KMS key policy to allow CloudTrail logs to be encrypted and decrypted. For more information, see Configure Amazon KMS key policies for CloudTrail (p. 387). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Using your own KMS key incurs Amazon KMS costs for encryption and decryption. After you associate an event data store with a KMS key, the KMS key cannot be removed or changed.

   Note
   To enable Amazon Key Management Service encryption for an organization event data store, you must use an existing KMS key for the management account.
7. (Optional) In the **Tags** section, you can add up to 50 tag key pairs to help you identify, sort, and control access to your event data store. For more information about how to use IAM policies to authorize access to an event data store based on tags, see Examples: Denying access to create or delete event data stores based on tags (p. 354). For more information about how you can use tags in Amazon, see Tagging Amazon resources in the *Amazon Web Services General Reference*.
8. Choose **Next** to configure the event data store.
9. On the **Choose events** page, choose **Events from integrations**.
10. From **Events from integration**, choose the source to deliver events to the event data store.
11. Provide a name to identify the integration's channel. The name can be 3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.
12. In **Resource policy**, configure the resource policy for the integration's channel. Resource policies are JSON policy documents that specify what actions a specified principal can perform on the resource and under what conditions. The accounts defined as principals in the resource policy can call the `PutAuditEvents` API to deliver events to your channel. The resource owner has implicit access to the resource if their IAM policy allows the `cloudtrail-data:PutAuditEvents` action.

    The information required for the policy is determined by the integration type. For a direction integration, CloudTrail automatically adds the partner's Amazon account IDs, and requires you to enter the unique external ID provided by the partner. For a solution integration, you must specify at least one Amazon account ID as principal, and can optionally enter an external ID to prevent against confused deputy.

    Note
    If you do not create a resource policy for the channel, only the channel owner can call the `PutAuditEvents` API on the channel.

    a. For a direct integration, enter the external ID provided by your partner. The integration partner provides a unique external ID, such as an account ID or a randomly generated string, to use for the integration to prevent against confused deputy. The partner is responsible for creating and providing a unique external ID.

       You can choose **How to find this?** to view the partner's documentation that describes how to find the external ID.

Note
If the resource policy includes an external ID, all calls to the `PutAuditEvents` API must include the external ID. However, if the policy does not define an external ID, the partner can still call the `PutAuditEvents` API and specify an `externalId` parameter.

b. For a solution integration, choose **Add Amazon account** to specify each Amazon account ID to add as a principal in the policy.

13. Choose **Next** to review your choices.
14. On the **Review and create** page, review your choices. Choose **Edit** to make changes to a section. When you're ready to create the event data store, choose **Create event data store**.
15. The new event data store is visible in the **Event data stores** table on the **Event data stores** page.
16. Provide the channel Amazon Resource Name (ARN) to the partner application. Instructions for providing the channel ARN to the partner application are found on the partner documentation website. For more information, choose the **Learn more** link for the partner on the **Available sources** tab of the **Integrations** page to open the partner's page in Amazon Web Services Marketplace.

The event data store starts ingesting partner events into CloudTrail through the integration's channel when you, the partner, or the partner applications calls the `PutAuditEvents` API on the channel.

Manage event data store lifecycles

The following are the lifecycle stages of an event data store.

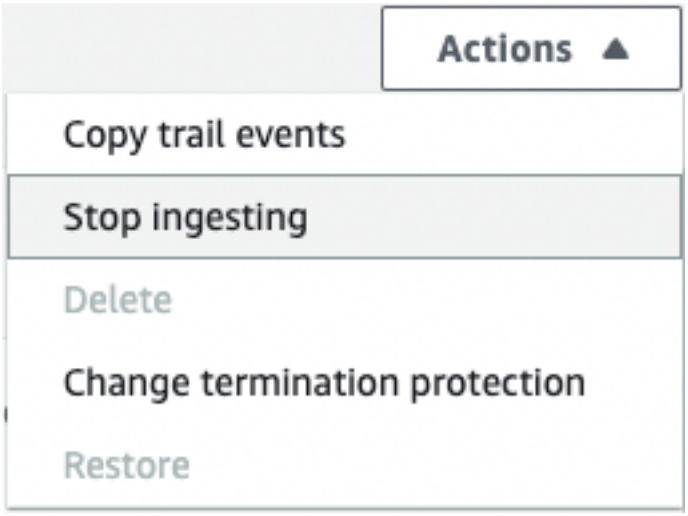
- `CREATED` – A short-term state indicating that the event data store has been created.
- `ENABLED` – The event data store is active and ingesting events. You can run queries and copy trail events to the event data store.
- `STARTING_INGESTION` – A short-term state indicating that the event data store will start ingesting live events.
- `STOPPING_INGESTION` – A short-term state indicating that the event data store will stop ingesting live events.
- `STOPPED_INGESTION` – The event data store is not ingesting live events. You can still run queries on any events already in the event data store and copy trail events to the event data store.
- `PENDING_DELETION` – The event data store was in an `ENABLED` or `STOPPED_INGESTION` state and has been deleted, but is within the seven-day wait period before permanent deletion. You cannot run queries on the event data store, and no operations can be performed on the event data store except restoration.

You can only delete an event data store if termination protection is disabled. Termination protection prevents an event data store from getting accidentally deleted. By default, termination protection is enabled on an event data store.

After you delete an event data store, it remains in the `PENDING_DELETION` state for seven days before it is permanently deleted. You can restore an event data store during the seven-day wait period. While in the `PENDING_DELETION` state, an event data store is not available for queries, and no other operations can be performed on the event data store except restore operations. An event data store that is pending deletion does not ingest events and does not incur costs.

Actions available on event data stores

To delete or restore an event data store, copy trail events, start or stop ingesting events, or turn on or turn off an event data store's termination protection, use commands on the **Actions** menu of the event data store's details page.



The option to **Copy trail events** is only available on event data stores that contain CloudTrail management and data events. The options to **Start ingestion** and **Stop ingestion** are only available on event data stores containing either CloudTrail events (management and data events), or Amazon Config configuration items.

Stop an event data store from ingesting events

By default, event data stores are configured to ingest events. You can stop an event data store from ingesting events by using the console, Amazon CLI, or APIs.

The options to **Start ingestion** and **Stop ingestion** are only available on event data stores containing either CloudTrail events (management and data events), or Amazon Config configuration items.

When you stop ingestion on an event data store, the event data store's state changes to `STOPPED_INGESTION`. You can still run queries on any events already in the event data store. You can also copy trail events to the event data store (if it contains only CloudTrail management or data events).

To stop an event data store from ingesting events

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, under **Lake**, choose **Event data stores**.
3. Choose the event data store.
4. From **Actions**, choose **Stop ingestion**.
5. When you are prompted to confirm, choose **Stop ingestion**. The event data store will stop ingesting live events.
6. To resume ingestion, choose **Start ingestion**.

Copy trail events to an event data store

You can copy trail events to a CloudTrail Lake event data store to create a point-in-time snapshot of events logged to the trail. Copying a trail's events does not interfere with the trail's ability to log events and does not modify the trail in any way.

You can copy trail events to an existing event data store configured for CloudTrail events, or you can create a new CloudTrail event data store and choose the **Copy trail events** option as part of event data

store creation. For more information about copying trail events to an existing event data store, see Copy trail events to an event data store (p. 178). For more information about creating a new event data store, see Create an event data store for CloudTrail events (p. 159).

If you are copying trail events to an organization event data store, you must use the management account for the organization. You cannot copy trail events using the delegated administrator account for an organization.

Scenarios

The following table describes some common scenarios for copying trail events and how you accomplish each scenario using the console.

Scenario	How do I accomplish this in the console?
Analyze and query historical trail events in CloudTrail Lake without ingesting new events	Create a new event data store and choose the **Copy trail events** option as part of event data store creation. When creating the event data store, deselect **Ingest events** (step 13 of the procedure) to ensure the event data store contains only the historical events for your trail and no future events.
Replace your existing trail with a CloudTrail Lake event data store	Create an event data store with the same event selectors as your trail to ensure that the event data store has the same coverage as your trail. To avoid duplicating events between the source trail and destination event data store, choose a date range for the copied events that is earlier than the creation of the event data store. After your event data store is created, you can turn off logging for the trail to avoid additional charges. For more information about pricing, see Amazon CloudTrail Pricing.

Topics

- Considerations for copying trail events (p. 174)
- Required permissions for copying trail events (p. 175)
- Copy trail events to an event data store (p. 178)
- Event copy details (p. 179)

Considerations for copying trail events

Consider the following factors when copying trail events.

- When copying trail events, CloudTrail uses the S3 `GetObject` API operation to retrieve the trail events in the source S3 bucket. There are some S3 archived storage classes, such as S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Outposts, and S3 Intelligent-Tiering Deep Archive tiers that are not accessible by using `GetObject`. To copy trail events stored in these archived storage classes, you must first restore a copy using the S3 `RestoreObject` operation. For information about restoring archived objects, see Restoring Archived Objects in the *Amazon S3 User Guide*.
- When you copy trail events to an event data store, CloudTrail copies all trail events regardless of the configuration of the destination event data store's event types, advanced event selectors, or Amazon Web Services Region.
- Before copying trail events, check the retention period of the event data store. CloudTrail only copies trail events that have an `eventTime` within the event data store's retention period. For example, if an

event data store's retention period is 90 days, then CloudTrail will not copy any trail events with an `eventTime` older than 90 days.

We recommend that when you choose a retention period, you consider both the age of the events that you want to copy as well as how long you want to keep the copied events in your event data store. For example, if you copy trail events that are 5 years old and specify a retention period of 7 years, the event data store will retain those events for two years.

- Before copying trail events, disable any access control lists (ACLs) attached to the source S3 bucket, and update the S3 bucket policy for the destination event data store. For more information about updating the S3 bucket policy, see Amazon S3 bucket policy for copying trail events (p. 142). For more information about disabling ACLs, see Controlling ownership of objects and disabling ACLs for your bucket in the *Amazon S3 User Guide*.
- CloudTrail only copies trail events from Gzip compressed log files that are in the source S3 bucket. CloudTrail does not copy trail events from uncompressed log files or log files that were compressed using a format other than Gzip.
- To avoid duplicating events between the source trail and destination event data store, choose a time range for the copied events that is earlier than the creation of the event data store.
- By default, CloudTrail only copies CloudTrail events contained in the S3 bucket's `CloudTrail` prefix and the prefixes inside the `CloudTrail` prefix, and does not check prefixes for other Amazon services. If you want to copy CloudTrail events contained in another prefix, you must choose the prefix when you copy trail events.
- To copy trail events to an organization event data store, you must use the management account for the organization. The delegated administrator account cannot copy trail events to an organization event data store.
- If you are copying trail events to an event data store for investigation and do not want to ingest any future events, you can stop ingestion on the event data store when you create it. When creating the event data store, deselect the **Ingest events** option (step 13 of the procedure (p. 160)) to ensure the event data store contains only the historical events for your trail and no future events.

Required permissions for copying trail events

Before copying trail events, ensure you have all the required permissions for your IAM role and the event data store's S3 bucket. You only need to update the IAM role permissions if you choose an existing IAM role to copy trail events. If you choose to create a new IAM role, CloudTrail provides all necessary permissions for the role.

If the source S3 bucket uses a KMS key for data encryption, ensure that the KMS key policy allows CloudTrail to decrypt data in the bucket. If the source S3 bucket uses multiple KMS keys, you must update each key's policy to allow CloudTrail to decrypt data in the bucket.

Topics

- IAM permissions for copying trail events (p. 175)
- Amazon S3 bucket policy for copying trail events (p. 177)
- KMS key policy for decrypting data in the source S3 bucket (p. 177)

IAM permissions for copying trail events

When copying trail events, you have the option to create a new IAM role, or use an existing IAM role. When you choose a new IAM role, CloudTrail creates an IAM role with the required permissions and no further action is required on your part.

If you choose an existing role, ensure the IAM role's policies allow CloudTrail to copy trail events to the destination event data store. This section provides examples of the required IAM role permission and trust policies.

The following example provides the permissions policy, which allows CloudTrail to copy trail events to the event data store's S3 bucket. Replace *myBucketName*, *myAccountID*, *region*, *prefix*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

Replace *key-region*, *keyAccountID*, and *keyID* with the values for the KMS key used to encrypt the source S3 bucket. You can omit the `AWSCloudTrailImportKeyAccess` statement if the source S3 bucket does not use a KMS key for encryption.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailImportBucketAccess",
      "Effect": "Allow",
      "Action": ["s3:ListBucket", "s3:GetBucketAcl"],
      "Resource": [
        "arn:aws:s3:::myBucketName"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
       }
    },
    {
      "Sid": "AWSCloudTrailImportObjectAccess",
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": [
        "arn:aws:s3:::myBucketName/prefix",
        "arn:aws:s3:::myBucketName/prefix/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
       }
    },
    {
      "Sid": "AWSCloudTrailImportKeyAccess",
      "Effect": "Allow",
      "Action": ["kms:GenerateDataKey","kms:Decrypt"],
      "Resource": [
        "arn:aws:kms:key-region:keyAccountID:key/keyID"
      ]
    }
  ]
}
```

The following example provides the IAM trust policy, which allows CloudTrail to assume an IAM role to copy trail events to the event data store's S3 bucket. Replace *myAccountID*, *region*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the

Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "myAccountID",
          "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
        }
      }
    }
  ]
}
```

Amazon S3 bucket policy for copying trail events

By default, Amazon S3 buckets and objects are private. Only the resource owner (the Amazon account that created the bucket) can access the bucket and objects it contains. The resource owner can grant access permissions to other resources and users by writing an access policy.

Before you copy trail events, you must update the S3 bucket policy for the destination event data store to allow CloudTrail to copy trail events to the bucket.

You can add the following statement to the event data store's S3 bucket policy to grant these permissions. Replace *roleArn* and *myBucketName* with the appropriate values for your configuration.

```
{
  "Sid": "AWSCloudTrailImportBucketAccess",
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetBucketAcl",
    "s3:GetObject"
  ],
  "Principal": {
    "AWS": "roleArn"
  },
  "Resource": [
    "arn:aws:s3:::myBucketName",
    "arn:aws:s3:::myBucketName/*"
  ]
},
```

KMS key policy for decrypting data in the source S3 bucket

If the source S3 bucket uses a KMS key for data encryption, ensure the KMS key policy provides CloudTrail with the `kms:Decrypt` and `kms:GenerateDataKey` permissions required to copy trail

events from an S3 bucket with SSE-KMS encryption enabled. If your source S3 bucket uses multiple KMS keys, you must update each key's policy. Updating the KMS key policy allows CloudTrail to decrypt data in the source S3 bucket, run validation checks to ensure that events conform to CloudTrail standards, and copy events into the CloudTrail Lake event data store.

The following example provides the KMS key policy, which allows CloudTrail to decrypt the data in the source S3 bucket. Replace *roleArn*, *myBucketName*, *myAccountID*, *region*, and *eventDataStoreArn* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail Lake, which may not be the same as the Amazon account ID for the S3 bucket.

```
{
  "Sid": "AWSCloudTrailImportDecrypt",
  "Effect": "Allow",
  "Action": [
          "kms:Decrypt",
          "kms:GenerateDataKey"
  ],
  "Principal": {
    "AWS": "roleArn"
  },
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::myBucketName/*"
    },
    "StringEquals": {
      "aws:SourceAccount": "myAccountID",
      "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:eventdataStore/eventDataStoreArn"
    }
  }
}
```

Copy trail events to an event data store

Use the following procedure to copy trail events to an existing event data store.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Event data stores**.
3. Choose **Copy trail events**.
4. On the **Copy trail events** page, for **Event source**, choose the trail that you want to copy. By default, CloudTrail only copies CloudTrail events contained in the S3 bucket's `CloudTrail` prefix and the prefixes inside the `CloudTrail` prefix, and does not check prefixes for other Amazon services. If you want to copy CloudTrail events contained in another prefix, choose **Enter S3 URI**, and then choose **Browse S3** to browse to the prefix. If the source S3 bucket for the trail uses a KMS key for data encryption, ensure that the KMS key policy allows CloudTrail to decrypt the data. If your source S3 bucket uses multiple KMS keys, you must update each key's policy to allow CloudTrail to decrypt the data in the bucket. For more information about updating the KMS key policy, see KMS key policy for decrypting data in the source S3 bucket (p. 177).
5. (Optional) For **Event source**, choose the time range for copying the events. If you choose a time range, CloudTrail checks the prefix and log file name to verify the name contains a date between the chosen start and end date before attempting to copy trail events. You can choose a **Relative range** or an **Absolute range**. To avoid duplicating events between the source trail and destination event data store, choose a time range that is earlier than the creation of the event data store.

- If you choose **Relative range**, you can choose to copy events logged in the last 6 months, 1 year, 2 years, 7 years, or a custom range. CloudTrail copies the events logged within the chosen time period.
- If you choose **Absolute range**, you can choose a specific start and end date. CloudTrail copies the events that occurred between the chosen start and end dates.

6. For **Delivery location**, choose the destination event data store from the drop-down list. The S3 bucket policy for the event data store must grant CloudTrail access to copy trail events into the bucket. For more information about updating the S3 bucket policy, see Amazon S3 bucket policy for copying trail events (p. 177).
7. For **Permissions**, choose from the following IAM role options. If you choose an existing IAM role, verify that the IAM role policy provides the necessary permissions. For more information about updating the IAM role permissions, see IAM permissions for copying trail events (p. 175).
   - Choose **Create a new role (recommended)** to create a new IAM role. For **Enter IAM role name**, enter a name for the role. CloudTrail automatically creates the necessary permissions for this new role.
   - Choose **Use a custom IAM role** to use a custom IAM role that is not listed. For **Enter IAM role ARN**, enter the IAM ARN.
   - Choose **Use an existing role** to choose an existing IAM role from the drop-down list.
8. Choose **Copy events**.
9. You are prompted to confirm. When you are ready to confirm, choose **Copy trail events to Lake**, and then choose **Copy events**.
10. On the **Copy details** page, you can see the copy status and review any failures. When a trail event copy completes, its **Copy status** is set to either **Completed** if there were no errors, or **Failed** if errors occurred.

    Note
    Details shown on the event copy details page are not in real-time. The actual values for details such as **Prefixes copied** may be higher than what is shown on the page. CloudTrail updates the details incrementally over the course of the event copy.
11. If the **Copy status** is **Failed**, fix any errors shown in **Copy failures**, and then choose **Retry copy**. When you retry a copy, CloudTrail resumes the copy at the location where the failure occurred.

For more information about viewing the details of a trail event copy, see Event copy details (p. 179).

Event copy details

After a trail event copy starts, you can view the event copy details, including the status of the copy, and information on any copy failures.

Note
Details shown on the event copy details page are not in real-time. The actual values for details such as **Prefixes copied** may be higher than what is shown on the page. CloudTrail updates the details incrementally over the course of the event copy.

To access the event copy details page

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the left navigation pane, under **Lake**, choose **Event data stores**.
3. Choose the event data store.
4. Choose the event copy in the **Event copy status** section.

Copy details

From **Copy details**, you can view the following details about the trail event copy.

- **Event log S3 location** - The location of the source S3 bucket containing the trail event log files.
- **Copy ID** - The ID for the copy.
- **Prefixes copied** - Represents the number of S3 prefixes copied. During a trail event copy, CloudTrail copies the events in the trail log files that are stored in the prefixes.
- **Copy status** - The status of the copy.
  - **Initializing** - Initial status shown when the trail event copy starts.
  - **In progress** - Indicates the trail event copy is in progress.

    Note
    You cannot copy trail events if another trail event copy is **In progress**. To stop a trail event copy, choose **Stop copy**.
  - **Stopped** - Indicates a **Stop copy** action occurred. To retry a trail event copy, choose **Retry copy**.
  - **Failed** - The copy completed, but some trail events failed to copy. Review the error messages in **Copy failures**. To retry a trail event copy, choose **Retry copy**. When you retry a copy, CloudTrail resumes the copy at the location where the failure occurred.
  - **Completed** - The copy completed without errors. You can query the copied trail events in the event data store.
- **Created time** - Indicates when the trail event copy started.
- **Finish time** - Indicates when the trail event copy completed or stopped.

Copy failures

From **Copy failures**, you can review the error location, error message, and error type for each copy failure. Common reasons for failure, include if an S3 prefix contained an uncompressed file, or contained a file delivered by a service other than CloudTrail. Another possible cause of failure relates to access issues. For example, if the event data store's S3 bucket did not grant CloudTrail access to import the events, you would get an `AccessDenied` error.

For each copy failure, review the following error information.

- The **Error location** - Indicates the location in the S3 bucket where the error occurred. If an error occurred because the source S3 bucket contained an uncompressed file, the **Error location** would include the prefix where you would find that file.
- The **Error message** - Provides an explanation for why the error occurred.
- The **Error type** - Provides the error type. For example, an **Error type** of `AccessDenied`, indicates that the error occurred because of a permissions issue. For more information about the required permissions for copying trail events, see Required permissions for copying trail events (p. 175).

After resolving any failures, choose **Retry copy**. When you retry a copy, CloudTrail resumes the copy at the location where the failure occurred.

Create an integration with an event source outside of Amazon

You can use CloudTrail to log and store user activity data from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers.

You can store, access, analyze, troubleshoot and take action on this data without maintaining multiple log aggregators and reporting tools.

Activity events from non-Amazon sources work by using *channels* to bring events into CloudTrail Lake from external partners that work with CloudTrail, or from your own sources. When you create a channel, you choose one or more event data stores to store events that arrive from the channel source. You can change the destination event data stores for a channel as needed, as long as the destination event data stores are set to log `eventCategory="ActivityAuditLog"` events. When you create a channel for events from an external partner, you provide a channel ARN to the partner or source application. The resource policy attached to the channel allows the source to transmit events through the channel. If a channel does not have a resource policy, only the channel owner can call the `PutAuditEvents` API on the channel.

CloudTrail has partnered with many event source providers, such as Okta and LaunchDarkly. When you create an integration with an event source outside Amazon, you can choose one of these partners as your event source, or choose **My custom integration** to integrate events from your own sources into CloudTrail. A maximum of one channel is allowed per source.

There are two types of integrations: direct and solution. With direct integrations, the partner calls the `PutAuditEvents` API to deliver events to the event data store for your Amazon account. With solution integrations, the application runs in your Amazon account and the application calls the `PutAuditEvents` API to deliver events to the event data store for your Amazon account.

From the **Integrations** page, you can choose the **Available sources** tab to the view the **Integration type** for partners.

To get started, create an integration to log events from partner or other application sources using the CloudTrail console.

Topics

- Create an integration with a CloudTrail partner (p. 181)
- Create a custom integration (p. 183)
- Additional information about integration partners (p. 185)
- CloudTrail Lake integrations event schema (p. 186)

Create an integration with a CloudTrail partner

When you create an integration with an event source outside Amazon, you can choose one of these partners as your event source. When you create an integration in CloudTrail with a partner application, the partner needs the Amazon Resource Name (ARN) of the channel that you create in this workflow to send events to CloudTrail. After you create the integration, you finish configuring the integration by following the partner's instructions to provide the required channel ARN to the partner. The integration starts ingesting partner events into CloudTrail after the partner calls `PutAuditEvents` on the integration's channel.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Integrations**.
3. On the **Add integration** page, enter a name for your channel. The name can be 3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.
4. Choose the partner application source from which you want to get events. If you're integrating with events from your own applications hosted on-premises or in the cloud, choose **My custom integration**.
5. From **Event delivery location**, choose to log the same activity events to existing event data stores, or create a new event data store.

If you choose to create a new event data store, enter a name for the event data store and specify the retention period in days. Valid values for the retention period are integers from 7 to 2557 (seven years). The event data store retains event data for the specified number of days. By default, event data is retained for 2557 days.

If you choose to log activity events to one or more existing event data stores, choose the event data stores from the list. The event data stores can only include activity events. The event type in the console must be **Events from integrations**. In the API, the `eventCategory` value must be `ActivityAuditLog`.

6. In **Resource policy**, configure the resource policy for the integration's channel. Resource policies are JSON policy documents that specify what actions a specified principal can perform on the resource and under what conditions. The accounts defined as principals in the resource policy can call the `PutAuditEvents` API to deliver events to your channel. The resource owner has implicit access to the resource if their IAM policy allows the `cloudtrail-data:PutAuditEvents` action.

   The information required for the policy is determined by the integration type. For a direction integration, CloudTrail automatically adds the partner's Amazon account IDs, and requires you to enter the unique external ID provided by the partner. For a solution integration, you must specify at least one Amazon account ID as principal, and can optionally enter an external ID to prevent against confused deputy.

   Note
   If you do not create a resource policy for the channel, only the channel owner can call the `PutAuditEvents` API on the channel.

   a. For a direct integration, enter the external ID provided by your partner. The integration partner provides a unique external ID, such as an account ID or a randomly generated string, to use for the integration to prevent against confused deputy. The partner is responsible for creating and providing a unique external ID.

      You can choose **How to find this?** to view the partner's documentation that describes how to find the external ID.

      Note
      If the resource policy includes an external ID, all calls to the `PutAuditEvents` API must include the external ID. However, if the policy does not define an external ID, the partner can still call the `PutAuditEvents` API and specify an `externalId` parameter.

   b. For a solution integration, choose **Add Amazon account** to specify an Amazon account ID to add as a principal in the policy.

7. (Optional) In the **Tags** area, you can add up to 50 tag key and value pairs to help you identify, sort, and control access to your event data store and channel. For more information about how to use IAM policies to authorize access to an event data store based on tags, see Examples: Denying access to create or delete event data stores based on tags (p. 354). For more information about how you can use tags in Amazon, see Tagging Amazon resources in the *Amazon Web Services General Reference*.

8. When you are ready to create the new integration, choose **Add integration**. There is no review page. CloudTrail creates the integration, but you must provide the channel Amazon Resource Name (ARN) to the partner application. Instructions for providing the channel ARN to the partner application are found on the partner documentation website. For more information, choose the **Learn more** link for the partner on the **Available sources** tab of the **Integrations** page to open the partner's page in Amazon Web Services Marketplace.

To finish the setup for your integration, provide the channel ARN to the partner or source application. Depending upon the integration type, either you, the partner, or the application runs the `PutAuditEvents` API to deliver activity events to the event data store for your Amazon account. After your activity events are delivered, you can use CloudTrail Lake to search, query, and analyze the data that

is logged from your applications. Your event data includes fields that match CloudTrail event payload, such as `eventVersion`, `eventSource`, and `userIdentity`.

Create a custom integration

You can use CloudTrail to log and store user activity data from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers. Perform the first half of this procedure in the CloudTrail Lake console, then call the `PutAuditEvents` API to ingest events, providing your channel ARN and event payload. After you use the `PutAuditEvents` API to ingest your application activity into CloudTrail, you can use CloudTrail Lake to search, query, and analyze the data that is logged from your applications.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Integrations**.
3. On the **Add integration** page, enter a name for your channel. The name can be 3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.
4. Choose **My custom integration**.
5. From **Event delivery location**, choose to log the same activity events to existing event data stores, or create a new event data store.

   If you choose to create a new event data store, enter a name for the event data store and specify the retention period in days. Valid values for the retention period are integers from 7 to 2557 (seven years). The event data store retains event data for the specified number of days. By default, event data is retained for 2557 days.

   If you choose to log activity events to one or more existing event data stores, choose the event data stores from the list. The event data stores can only include activity events. The event type in the console must be **Events from integrations**. In the API, the `eventCategory` value must be `ActivityAuditLog`.
6. In **Resource policy**, configure the resource policy for the integration's channel. Resource policies are JSON policy documents that specify what actions a specified principal can perform on the resource and under what conditions. The accounts defined as principals in the resource policy can call the `PutAuditEvents` API to deliver events to your channel.

   Note
   If you do not create a resource policy for the channel, only the channel owner can call the `PutAuditEvents` API on the channel.

   a. (Optional) Enter a unique external ID to provide an extra layer of protection. The external ID is a unique string such as an account ID or a randomly generated string, to prevent against confused deputy.

      Note
      If the resource policy includes an external ID, all calls to the `PutAuditEvents` API must include the external ID. However, if the policy does not define an external ID, you can still call the `PutAuditEvents` API and specify an `externalId` parameter.

   b. Choose **Add Amazon account** to specify each Amazon account ID to add as a principal in the resource policy for the channel.
7. (Optional) In the **Tags** area, you can add up to 50 tag key and value pairs to help you identify, sort, and control access to your event data store and channel. For more information about how to use IAM policies to authorize access to an event data store based on tags, see Examples: Denying access to create or delete event data stores based on tags (p. 354). For more information about how you can use tags in Amazon, see Tagging Amazon resources in the *Amazon Web Services General Reference*.

8. When you are ready to create the new integration, choose **Add integration**. There is no review page. CloudTrail creates the integration, but to integrate your custom events, you must specify the channel ARN in a `PutAuditEvents` request.
9. Call the `PutAuditEvents` API to ingest your activity events into CloudTrail. You can add up to 100 activity events (or up to 1 MB) per `PutAuditEvents` request. You'll need the channel ARN that you created in preceding steps, the payload of events that you want CloudTrail to add, and the external ID (if specified for your resource policy). Be sure that there is no sensitive or personally-identifying information in event payload before ingesting it into CloudTrail. Events that you ingest into CloudTrail must follow the CloudTrail Lake integrations event schema (p. 186).

   Tip
   Use Amazon CloudShell to be sure you are running the most current Amazon APIs.

   The following examples show how to use the **put-audit-events** CLI command. The **--audit-events** and **--channel-arn** parameters are required. You need the ARN of the channel that you created in the preceding steps, which you can copy from the integration details page. The value of **--audit-events** is a JSON array of event objects. `--audit-events` includes a required ID from the event, the required payload of the event as the value of `EventData`, and an optional checksum (p. 185) to help validate the integrity of the event after ingestion into CloudTrail.

```
aws cloudtrail-data put-audit-events \
--region region \
--channel-arn $ChannelArn \
--audit-events \
id="event_ID",eventData='"{event_payload}"' \
id="event_ID",eventData='"{event_payload}"',eventDataChecksum="optional_checksum"
```

   The following is an example command with two event examples.

```
aws cloudtrail-data put-audit-events \
--region us-east-1 \
--channel-arn arn:aws:cloudtrail:us-east-1:01234567890:channel/EXAMPLE8-0558-4f7e-
a06a-43969EXAMPLE \
--audit-events \
id="EXAMPLE3-0f1f-4a85-9664-d50a3EXAMPLE",eventData='"{\"eventVersion\":\0.01\",
\"eventSource\":\"custom1.domain.com\", ...
\}"' \
id="EXAMPLE7-a999-486d-b241-b33a1EXAMPLE",eventData='"{\"eventVersion\":\0.02\",
\"eventSource\":\"custom2.domain.com\", ...
\}"',eventDataChecksum="EXAMPLE6e7dd61f3ead...93a691d8EXAMPLE"
```

   The following example command adds the `--cli-input-json` parameter to specify a JSON file (`custom-events.json`) of event payload.

```
aws cloudtrail-data put-audit-events \
--channel-arn $channelArn \
--cli-input-json file://custom-events.json \
--region us-east-1
```

   The following are the sample contents of the example JSON file, `custom-events.json`.

```
{
    "auditEvents": [
      {
        "eventData": "{\"version\":\"eventData.version\",\"UID\":\"UID\",
        \"userIdentity\":{\"type\":\"CustomUserIdentity\",\"principalId\":\"principalId
\",
```

```
        \"details\":{\"key\":\"value\"}},\"eventTime\":\"2021-10-27T12:13:14Z\",
\"eventName\":\"eventName\",
        \"userAgent\":\"userAgent\",\"eventSource\":\"eventSource\",
        \"requestParameters\":{\"key\":\"value\"},\"responseElements\":{\"key\":\"value
\"},
        \"additionalEventData\":{\"key\":\"value\"},
        \"sourceIPAddress\":\"source_IP_address\",\"recipientAccountId\":
\"recipient_account_ID\"}",
        "id": "1"
      }
   ]
}
```

(Optional) Calculate a checksum value

The checksum that you specify as the value of `EventDataChecksum` in a `PutAuditEvents` request helps you verify that CloudTrail receives the event that matches with the checksum; it helps verify the integrity of events. The checksum value is a base64-SHA256 algorithm that you calculate by running the following command.

```
printf %s "{"eventData": "{\"version\":\"eventData.version\",\"UID\":\"UID\",
        \"userIdentity\":{\"type\":\"CustomUserIdentity\",\"principalId\":\"principalId\",
        \"details\":{\"key\":\"value\"}},\"eventTime\":\"2021-10-27T12:13:14Z\",\"eventName
\":\"eventName\",
        \"userAgent\":\"userAgent\",\"eventSource\":\"eventSource\",
        \"requestParameters\":{\"key\":\"value\"},\"responseElements\":{\"key\":\"value\"},
        \"additionalEventData\":{\"key\":\"value\"},
        \"sourceIPAddress\":\"source_IP_address\",
        \"recipientAccountId\":\"recipient_account_ID\"}",
        "id": "1"}" \
 | openssl dgst -binary -sha256 | base64
```

The command returns the checksum. The following is an example.

```
EXAMPLEHjkI8iehvCUCWTIAbNYkOgO/t0YNw+7rrQE=
```

The checksum value becomes the value of `EventDataChecksum` in your `PutAuditEvents` request. If the checksum doesn't match with the one for the provided event, CloudTrail rejects the event with an `InvalidChecksum` error.

Additional information about integration partners

The table in this section provides the source name for each integration partner and identifies the integration type (direct or solution).

The information in the **Source name** column is required when calling the `CreateChannel` API. You specify the source name as the value for the `Source` parameter.

Partner name (console)	Source name (API)	Integration type
My custom integration	`Custom`	solution
Cloud Storage Security	`CloudStorageSecurityConsole`	solution
Clumio	`Clumio`	direct
CrowdStrike	`CrowdStrike`	solution

Partner name (console)	Source name (API)	Integration type
CyberArk	`CyberArk`	solution
GitHub	`GitHub`	solution
Kong Inc	`KongGatewayEnterprise`	solution
LaunchDarkly	`LaunchDarkly`	direct
Netskope	`NetskopeCloudExchange`	solution
Nordcloud, an IBM Company	`IBMMulticloud`	direct
MontyCloud	`MontyCloud`	direct
Okta	`OktaSystemLogEvents`	solution
One Identity	`OneLogin`	solution
Shoreline.io	`Shoreline`	solution
Snyk.io	`Snyk`	direct
Wiz	`WizAuditLogs`	solution

View partner documentation

You can learn more about a partner's integration with CloudTrail Lake by viewing their documentation.

To view partner documentation

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Integrations**.
3. From the **Integrations** page, choose **Available sources**, then choose **Learn more** for the partner whose documentation you want to view.

CloudTrail Lake integrations event schema

The following table describes the required and optional schema elements that match those in CloudTrail event records. The contents of `eventData` are provided by your events; other fields are provided by CloudTrail after ingestion.

CloudTrail event record contents are described in more detail in CloudTrail record contents (p. 402).

- Fields that are provided by CloudTrail after ingestion (p. 186)
- Fields that are provided by your events (p. 188)

Fields that are provided by CloudTrail after ingestion

Field name	Input type	Requirement	Description
eventVersion	string	Required	The event version.

Field name	Input type	Requirement	Description
eventCategory	string	Required	The event category. For non-Amazon events, the value is `ActivityAuditLog`.
eventType	string	Required	The event type. For non-Amazon events, the valid value is `ActivityLog`.
eventID	string	Required	A unique ID for an event.
eventTime	string	Required	Event timestamp, in `yyyy-MM-DDTHH:mm:ss` format, in Universal Coordinated Time (UTC).
awsRegion	string	Required	The Amazon Web Services Region where the `PutAuditEvents` call was made.
recipientAccountId		Required	Represents the account ID that received this event. CloudTrail populates this field by calculating it from event payload.
addendum	-	Optional	Shows information about why event processing was delayed. If information was missing from an existing event, the addendum block includes the missing information and a reason for why it was missing.
• reason	string	Optional	The reason that the event or some of its contents were missing.
• updatedFields	string	Optional	The event record fields that are updated by the addendum. This is only provided if the reason is `UPDATED_DATA`.

Field name	Input type	Requirement	Description
• originalUID	string	Optional	The original event UID from the source. This is only provided if the reason is `UPDATED_DATA`.
• originalEventID	string	Optional	The original event ID. This is only provided if the reason is `UPDATED_DATA`.
metadata	-	Required	Information about the channel that the event used.
• ingestionTime	string	Required	The timestamp when the event was processed, in `yyyy-MM-DDTHH:mm:ss` format, in Universal Coordinated Time (UTC).
• channelARN	string	Required	The ARN of the channel that the event used.

Fields that are provided by customer events

Field name	Input type	Requirement	Description
eventData	-	Required	The audit data sent to CloudTrail in a `PutAuditEvents` call.
• version	string	Required	The version of the event from its source. Length constraints: Maximum length of 256.
• userIdentity	-	Required	Information about the user who made a request.
• • type	string	Required	The type of user identity. Length constraints: Maximum length of 128.
• • principalId	string	Required	A unique identifier for the actor of the event.

Field name	Input type	Requirement	Description
			Length constraints: Maximum length of 1024.
• • details	JSON object	Optional	Additional information about the identity.
• userAgent	string	Optional	The agent through which the request was made. Length constraints: Maximum length of 1024.
• eventSource	string	Required	This is the partner event source, or the custom application about which events are logged. Length constraints: Maximum length of 1024.
• eventName	string	Required	The requested action, one of the actions in the API for the source service or application. Length constraints: Maximum length of 1024.
• eventTime	string	Required	Event timestamp, in `yyyy-MM-DDTHH:mm:ss` format, in Universal Coordinated Time (UTC).
• UID	string	Required	The UID value that identifies the request. The service or application that is called generates this value. Length constraints: Maximum length of 1024.

Field name	Input type	Requirement	Description
• requestParameters	JSON object	Optional	The parameters, if any, that were sent with the request. This field has a maximum size of 100 kB, and content exceeding the limit is rejected.
• responseElements	JSON object	Optional	The response element for actions that make changes (create, update, or delete actions). This field has a maximum size of 100 kB, and content exceeding the limit is rejected.
• errorCode	string	Optional	A string representing an error for the event. Length constraints: Maximum length of 256.
• errorMessage	string	Optional	The description of the error. Length constraints: Maximum length of 256.
• sourceIPAddress	string	Optional	The IP address from which the request was made. Both IPv4 and IPv6 addresses are accepted.
• recipientAccountId	string	Required	Represents the account ID that received this event. The account ID must be the same as the Amazon account ID that owns the channel.
• additionalEventData	JSON object	Optional	Additional data about the event that was not part of the request or response. This field has a maximum size of 28 kB, and content exceeding that limit is rejected.

The following example shows the hierarchy of schema elements that match those in CloudTrail event records.

```
{
    "eventVersion": String,
    "eventCategory": String,
    "eventType": String,
    "eventID": String,
    "eventTime": String,
    "awsRegion": String,
    "recipientAccountId": String,
    "addendum": {
       "reason": String,
       "updatedFields": String,
       "originalUID": String,
       "originalEventID": String
    },
    "metadata" : {
       "ingestionTime": String,
       "channelARN": String
    },
    "eventData": {
        "version": String,
        "userIdentity": {
          "type": String,
          "principalId": String,
          "details": {
             JSON
          }
        },
        "userAgent": String,
        "eventSource": String,
        "eventName": String,
        "eventTime": String,
        "UID": String,
        "requestParameters": {
           JSON
        },
        "responseElements": {
           JSON
        },
        "errorCode": String,
        "errorMessage": String,
        "sourceIPAddress": String,
        "recipientAccountId": String,
        "additionalEventData": {
           JSON
        }
    }
}
```

View Lake dashboards

You can use CloudTrail Lake dashboards to visualize the events in an event data store. You can select from several different dashboard types. The dashboard types available for an event data store are dependent upon the advanced event selectors configuration of the event data store. For example, if a dashboard type displays information about CloudTrail management events, you can only select the dashboard if the currently selected event data store collects CloudTrail management events.

Each dashboard type consists of multiple widgets and each widget represents a SQL query. To view the query for a widget, choose **View and analyze in query editor** to open up the query editor. You can't

modify the system-generated query that is used to populate the widget, but you can make edits to the query and run the query in the query editor for further analysis.

To populate and update a dashboard, choose **Run queries**. When you choose **Run queries**, CloudTrail runs system-generated queries on your behalf. Because running queries incur costs, CloudTrail asks you to acknowledge the costs associated with running queries. This is a one time confirmation. For more information about CloudTrail pricing, see CloudTrail Pricing.

Topics

- Limitations (p. 192)
- Prerequisites (p. 192)
- Choosing a dashboard (p. 192)
- Filtering a dashboard on a date or time range (p. 193)
- Viewing the query for a dashboard widget (p. 193)

Limitations

The following limitations apply to the current release.

- The current release doesn't support customized dashboards, widgets, or queries.
- The current release only provides dashboards for event data stores that collect CloudTrail events (data events, management events).
- The current release doesn't support editing the system-generated queries used to populate the dashboard. You can view and edit the underlying query for any widget on the **Query Editor** tab, however, any changes you make to the query are intended for supplemental analysis outside of the dashboard.

Prerequisites

The following prerequisites apply to Lake dashboards.

- To view and use Lake dashboards, you must create at least one CloudTrail Lake event data store. You can create event data stores using the console, Amazon CLI, or SDKs. For information about creating an event data store using the console, see Create an event data store for CloudTrail events (p. 159). For information about creating an event data store using the Amazon CLI, see Managing CloudTrail Lake by using the Amazon CLI (p. 214).
- To populate the dashboard, CloudTrail runs queries on your behalf. The first time you view the **Dashboards** page, CloudTrail asks you to acknowledge the costs associated with running queries. Choose **I agree** to acknowledge the cost of running queries.

Choosing a dashboard

Use the following procedure to choose an event data store and dashboard type to view.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the left navigation pane, under **Lake**, choose **Dashboard**.
3. Choose the event data store for which you want to visualize data.
4. Choose the dashboard type you want to view. The dashboards list is populated based upon the advanced event selectors configuration of the selected event data store.

The following are the possible dashboard types.

- **Overview** dashboard - Shows the most active users, Amazon Web Services Regions, and Amazon Web Services by event count. You can also view information about `read` and `write` management event activity, most throttled events, and the top errors. This dashboard is available for event data stores that collect management events.
- **Management Events** dashboard - Shows console sign-in events, access denied events, destructive actions, and top errors by user. You can also view information about TLS versions and outdated TLS calls by user. This dashboard is available for event data stores that collect management events.
- **S3 Data Events** dashboard - Shows S3 account activity, most accessed S3 objects, top S3 users, and top S3 actions. This dashboard is available for event data stores that collect Amazon S3 data events.

5. Choose to filter the dashboard data by an **Absolute range** or **Relative range**. Choose **Absolute range** to select a specific date and time range. Choose **Relative range** to select a predefined time range or a custom range. By default, the dashboard displays event data for the past 24 hours.
6. Choose **Run queries** to run the queries for the dashboard's widgets.

Filtering a dashboard on a date or time range

By default, the dashboard displays data for the past 24 hours. You can filter a dashboard by an **Absolute range** or **Relative range**.

Choose **Absolute range** to select a specific date and time range.

Choose **Relative range** to select a predefined time range or a custom range.

After you've chosen the time range, choose **Run queries** to refresh the dashboard.

Viewing the query for a dashboard widget

Each widget represents a SQL query. To view the query for a widget, choose **View and analyze in query editor** to open up the query editor. Using the query editor, you can further refine the query outside the dashboard and run the query to see the results of your updated query. For more information about working with queries, see Create or edit a query (p. 193).

Note
You cannot modify the system-generated query for a dashboard widget. Any changes made to the query on the **Query Editor** tab are intended solely for further analysis outside of the dashboard.

Create or edit a query

Queries in CloudTrail are authored in SQL. You can build a query on the CloudTrail Lake **Editor** tab by writing the query in SQL from scratch, or by opening a saved or sample query and editing it. You cannot overwrite an included sample query with your changes, but you can save it as a new query. For more information about the SQL query language that is allowed, see CloudTrail Lake SQL constraints (p. 227).

An unbounded query (such as `SELECT * FROM edsID`) scans all data in your event data store. To help control costs, we recommend that you constrain queries by adding starting and ending `eventTime` time

stamps to queries. The following is an example that searches for all events in a specified event data store where the event time is after (>) January 5, 2022 at 1:51 p.m. and before (<) January 19, 2022 at 1:51 p.m. Because an event data store has a minimum retention period of seven days, the minimum time span between starting and ending `eventTime` values is also seven days.

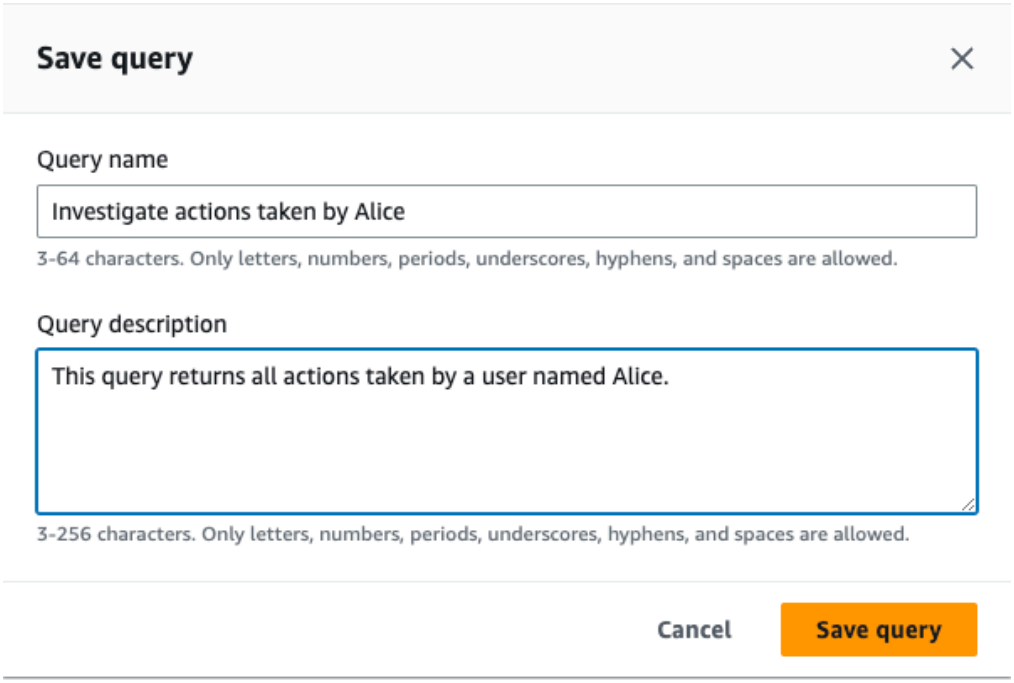
```
SELECT *
FROM eds-ID
WHERE
     eventtime >='2022-01-05 13:51:00' and eventtime < ='2022-01-19 13:51:00'
```

In this walkthrough, we open one of the sample queries, edit it to find actions taken by a specific user named `Alice`, and save it as a new query. You can also edit a saved query on the **Saved queries** tab, if you have saved queries.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Query**.
3. On the **Query** page, choose the **Sample queries** tab.
4. Open a sample query by choosing the **Query name**. This opens the query in the **Editor** tab. In this example, we'll select the query named **Investigate user actions** and edit the query to find the actions for a specific user named `Alice`.
5. In the **Editor** tab, edit the `WHERE` line to specify the user that you want to investigate and update the `eventTime` values as needed. The value of `FROM` is the ID portion of the event data store's ARN and is automatically populated by CloudTrail when you choose the event data store.

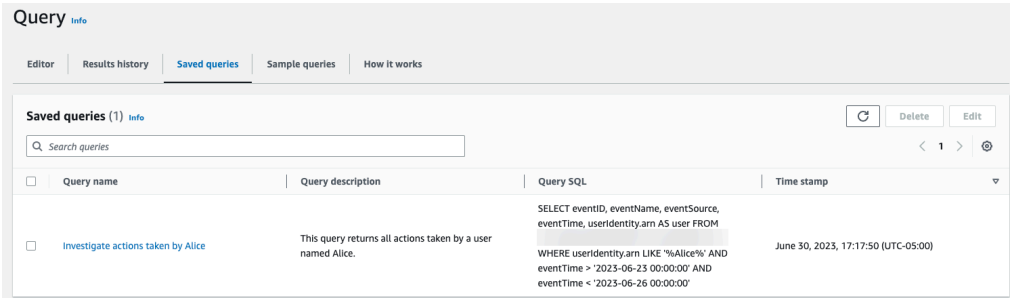
   ```
   SELECT
       eventID, eventName, eventSource, eventTime, userIdentity.arn AS user
   FROM
       event-data-store-id
   WHERE
       userIdentity.arn LIKE '%Alice%'
       AND eventTime > '2023-06-23 00:00:00' AND eventTime < '2023-06-26 00:00:00'
   ```

6. You can run a query before you save it, to verify that the query works. To run a query, choose an event data store from the **Event data store** drop-down list, and then choose **Run**. View the **Status** column of the **Command output** tab for the active query to verify that a query ran successfully.
7. When you have updated the sample query, choose **Save**.
8. In **Save query**, enter a name and description for the query. Choose **Save query** to save your changes as the new query. To discard changes to a query, choose **Cancel**, or close the **Save query** window.



Note
Saved queries are tied to your browser; if you use a different browser or a different device to access the CloudTrail console, the saved queries are not available.

9. Open the **Saved queries** tab to see the new query in the table.



Query editor tools

A toolbar at the upper right of the query editor offers commands to help author and format your SQL query.



The following list describes the commands on the toolbar.

- **Undo** – Reverts the last content change made in the query editor.
- **Redo** – Repeats the last content change made in the query editor.
- **Format selected** – Arranges the query editor content according to SQL formatting and spacing conventions.

- **Comment/uncomment selected** - Comments the selected portion of the query if it is not already commented. If the selected portion is already commented, choosing this option removes the comment.

Example queries

This section describes how you can access sample queries in the CloudTrail console and includes a few example CloudTrail Lake queries to help you get started.

> **Note**
> You can also view queries created by the GitHub community. For more information and to view these sample queries, see CloudTrail Lake sample queries on the GitHub website. Amazon CloudTrail has not evaluated the queries in GitHub.

Topics

- Viewing sample queries in the CloudTrail console (p. 196)
- Example: Find all principal user identities who called CreateBucket on January 22, 2023 (p. 196)
- Example: Find all APIs that a user called on January 22, 2023 (p. 197)
- Example: Find the number of API calls since January 1, 2023, grouped by eventName and eventSource (p. 198)
- Example: Find all users who signed in to the console in a set of Regions (p. 199)
- Example: Find all CloudTrail Lake queries that were run in January 2023 (p. 200)

Viewing sample queries in the CloudTrail console

The CloudTrail console provides a number of sample queries that can help you get started writing your own queries.

To access the sample queries in the CloudTrail console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Query**.
3. Choose the **Sample queries** tab.
4. To edit a sample query, choose the **Query name**. For information about running queries, see Run a query and save query results (p. 200).

Example: Find all principal user identities who called `CreateBucket` on January 22, 2023

```
SELECT
    userIdentity.principalid,
    eventName
FROM
    event_data_store_ID
WHERE
    userIdentity.principalid IS NOT NULL
AND
    eventTime > '2023-01-22 00:00:00'
AND
    eventTime < '2023-01-23 00:00:00'
AND
```

```
    eventName='CreateBucket'
```

Results

```
{
    "QueryStatus": "FINISHED",
    "QueryStatistics": {
        "ResultsCount": 1,
        "TotalResultsCount": 1,
        "BytesScanned": 25077
    },
    "QueryResultRows": [
        [
            {
                "principalid": "principal_ID"
            },
            {
                "eventName": "CreateBucket"
            }
        ]
    ]
}
```

Example: Find all APIs that a user called on January 22, 2023

```
SELECT
    eventID,
    eventName,
    eventSource,
    eventTime
FROM
    event_data_store_ID
WHERE
    userIdentity.username = 'bob'
AND
    eventTime > '2023-01-22 00:00:00'
AND
    eventTime < '2023-01-23 00:00:00'
```

Results

```
{
    "QueryStatus": "FINISHED",
    "QueryStatistics": {
        "ResultsCount": 2,
        "TotalResultsCount": 2,
        "BytesScanned": 13287
    },
    "QueryResultRows": [
        [
            {
                "eventID": "EXAMPLE-c3b6-43e4-aa35-b2490EXAMPLE"
            },
            {
                "eventName": "DescribeQuery"
            },
            {
                "eventSource": "cloudtrail.amazonaws.com"
            },
            {
                "eventTime": "2023-01-22 16:53:53.000"
            }
```

```
        ],
        [
            {
                "eventID": "EXAMPLE6-ac95-4b37-b587-76a80EXAMPLE"
            },
            {
                "eventName": "ListBuckets"
            },
            {
                "eventSource": "s3.amazonaws.com"
            },
            {
                "eventTime": "2023-01-22 20:25:01.000"
            }
        ]
    ]
}
```

Example: Find the number of API calls since January 1, 2023, grouped by eventName and eventSource

```
SELECT
    eventSource,
    eventName,
    COUNT(*) AS apiCount
FROM
    event_data_store_ID
WHERE
    eventTime > '2023-01-01 00:00:00'
GROUP BY
    eventSource, eventName
ORDER BY
    apiCount DESC
```

Results

```
{
    "QueryStatus": "FINISHED",
    "QueryStatistics": {
        "ResultsCount": 3,
        "TotalResultsCount": 3,
        "BytesScanned": 10442
    },
    "QueryResultRows": [
        [
            {
                "eventSource": "s3.amazonaws.com"
            },
            {
                "eventName": "PutObject"
            },
            {
                "apiCount": "96059"
            }
        ],
        [
            {
                "eventSource": "dynamodb.amazonaws.com"
            },
            {
                "eventName": "DescribeTable"
            },
```

```
            {
                "apiCount": "49426"
            }
        ],
        [
            {
                "eventSource": "sts.amazonaws.com"
            },
            {
                "eventName": "AssumeRole"
            },
            {
                "apiCount": "45617"
            }
        ]
    ]
}
```

Example: Find all users who signed in to the console in a set of Regions

```
SELECT
    eventTime,
    useridentity.arn,
    awsRegion
FROM
    event_data_store_ID
WHERE
    awsRegion in ('us-east-1', 'us-west-2')
AND
    eventName = 'ConsoleLogin'
```

Results

```
{
    "QueryStatus": "FINISHED",
    "QueryStatistics": {
        "ResultsCount": 2,
        "TotalResultsCount": 2,
        "BytesScanned": 15580
    },
    "QueryResultRows": [
        [
            {
                "eventTime": "2022-02-08 19:54:44.000"
            },
            {
                "arn": "arn:aws:sts::123456789012:assumed-role/example-identity"
            },
            {
                "awsRegion": "us-east-1"
            }
        ],
        [
            {
                "eventTime": "2022-01-21 16:38:27.000"
            },
            {
                "arn": "arn:aws:sts::123456789012:assumed-role/example-identity"
            },
            {
                "awsRegion": "us-west-2"
```

```
            }
        ]
    ]
}
```

Example: Find all CloudTrail Lake queries that were run in January 2023

```
SELECT
    element_at(responseElements, 'queryId'),
    element_at(requestParameters, 'queryStatement')
FROM
    event_data_store_ID
WHERE
    eventName='StartQuery'
AND
    eventSource = 'cloudtrail.amazonaws.com'
AND
    responseElements IS NOT NULL
AND
    eventTime > '2023-01-01 00:00:00'
AND
    eventTime < '2023-02-01 00:00:00'
```

Results

```
{
    "QueryStatus": "FINISHED",
    "QueryStatistics": {
        "ResultsCount": 1,
        "TotalResultsCount": 1,
        "BytesScanned": 13002
    },
    "QueryResultRows": [
        [
            {
                "_col0": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE"
            },
            {
                "_col1": "select * from event_data_store_ID limit 1;"
            }
        ]
    ]
}
```

Run a query and save query results

After you choose or save a query, you can run a query on an event data store.

When you run a query, you have the option to save the query results to an Amazon S3 bucket. When you run queries in CloudTrail Lake, you incur charges based on the amount of data scanned by the query. There are no additional CloudTrail Lake charges for saving query results to an S3 bucket, however, there are S3 storage charges. For more information about S3 pricing, see Amazon S3 pricing.

When you save query results, the query results may display in the CloudTrail console before they are viewable in the S3 bucket since CloudTrail delivers the query results after the query scan completes. While most queries complete within a few minutes, depending on the size of your event data store, it can take considerably longer for CloudTrail to deliver query results to your S3 bucket. CloudTrail delivers the

query results to the S3 bucket in compressed gzip format. On average, after the query scan completes you can expect a latency of 60 to 90 seconds for every GB of data delivered to the S3 bucket.

To run a query using CloudTrail Lake

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. From the navigation pane, under **Lake**, choose **Query**.
3. On the **Saved queries** or **Sample queries** tabs, choose a query to run by choosing the **Query name**.
4. On the **Editor** tab, for **Event data store**, choose an event data store from the drop-down list.
5. (Optional) On the **Editor** tab, choose **Save results to S3** to save the query results to an S3 bucket. When you choose the default S3 bucket, CloudTrail creates and applies the required bucket policies. For more information about saving query results, see Additional information about saved query results (p. 201).

   Note
   To use a different bucket, specify a bucket name, or choose **Browse S3** to choose a bucket. The bucket policy must grant CloudTrail permission to deliver query results to the bucket. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail Lake query results (p. 368).
6. On the **Editor** tab, choose **Run**.

   Depending on the size of your event data store, and the number of days of data it includes, a query can take several minutes to run. The **Command output** tab shows the status of a query, and whether a query is finished running. When a query has finished running, open the **Query results** tab to see a table of results for the active query (the query currently shown in the editor).

   Note
   Queries that run for longer than one hour might time out. You can still get partial results that were processed before the query timed out. CloudTrail does not deliver partial query results to an S3 bucket. To avoid a time out, you can refine your query to limit the amount of data scanned by specifying a narrower time range.

Additional information about saved query results

After you save query results, you can download the saved query results from the S3 bucket. For more information about finding and downloading saved query results, see Get and download saved query results (p. 202).

You can also validate saved query results to determine whether the query results were modified, deleted, or unchanged after CloudTrail delivered the query results. For more information about validating saved query results, see Validate saved query results (p. 204).

View query results

After your query finishes, you can view its results. The results of a query are available for seven days after the query finishes. You can view results for the active query on the **Query results** tab, or you can access results for all recent queries on the **Results history** tab on the **Lake** home page.

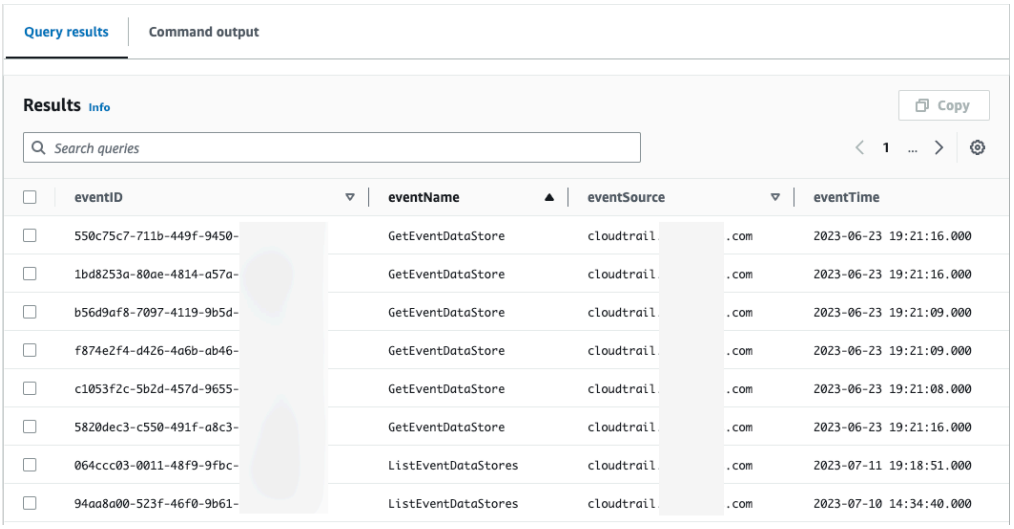
Query results can change from older runs of a query to newer ones, as later events in the query period can be logged between queries.

When you save query results, the query results may display in the CloudTrail console before they are viewable in the S3 bucket since CloudTrail delivers the query results after the query scan completes. While most queries complete within a few minutes, depending on the size of your event data store, it can take considerably longer for CloudTrail to deliver query results to your S3 bucket. CloudTrail delivers the query results to the S3 bucket in compressed gzip format.

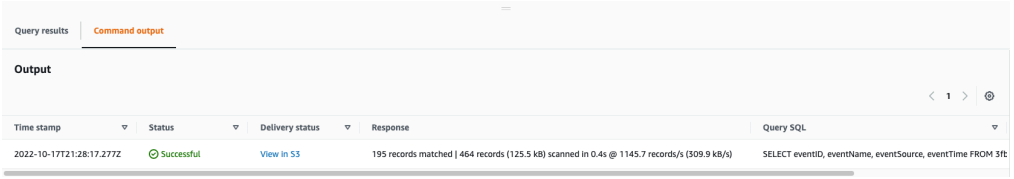
On average, after the query scan completes you can expect a latency of 60 to 90 seconds for every GB of data delivered to the S3 bucket. For more information about finding and downloading saved query results, see Get and download saved query results (p. 202).

Note
Queries that run for longer than one hour might time out. You can still get partial results that were processed before the query timed out. CloudTrail does not deliver partial query results to an S3 bucket. To avoid a time out, you can refine your query to limit the amount of data scanned by specifying a narrower time range.

1. On the **Query results** tab for an active query, each row represents an event result that matched the query. Filter results by entering all or part of an event field value in the search bar. To copy an event, choose the event you want to copy and then choose **Copy**.



2. On the **Command output** tab, view metadata about the query that was run, such as the event data store ID, run time, number of results scanned, and whether or not the query was successful. If you saved the query results to an Amazon S3 bucket, the metadata also includes a link to the S3 bucket containing the saved query results.



Get and download saved query results

After you save query results, you need to be able to locate the file containing the query results. CloudTrail delivers your query results to an Amazon S3 bucket that you specify when you save the query results.

Note
When you save query results, the query results may display in the console before they are viewable in the S3 bucket since CloudTrail delivers the query results after the query scan completes. While most queries complete within a few minutes, depending on the size of your event data store, it can take considerably longer for CloudTrail to deliver query results to your S3 bucket. CloudTrail delivers the query results to the S3 bucket in compressed gzip format. On average, after the query scan completes you can expect a latency of 60 to 90 seconds for every GB of data delivered to the S3 bucket.

Topics

- Find your CloudTrail Lake saved query results (p. 203)
- Download your CloudTrail Lake saved query results (p. 203)

Find your CloudTrail Lake saved query results

CloudTrail publishes query result and sign files to your S3 bucket. The query result file contains the output of the saved query and the sign file provides the signature and hash value for the query results. You can use the sign file to validate the query results. For more information about validating query results, see Validate saved query results (p. 204).

To retrieve a query result or sign file, you can use the Amazon S3 console, the Amazon S3 command line interface (CLI), or the API.

To find your query results and sign files with the Amazon S3 console

1. Open the Amazon S3 console.
2. Choose the bucket you specified.
3. Navigate through the object hierarchy until you find the query result and sign files. The query result file has a .csv.gz extension and the sign file has a .json extension.

You will navigate through an object hierarchy that is similar to the following example, but with a different bucket name, account ID, date, and query ID.

```
All Buckets
    Bucket_Name
        AWSLogs
            Account_ID;
                CloudTrail-Lake
                    Query
                        2022
                            06
                              20
                                Query_ID
```

Download your CloudTrail Lake saved query results

When you save query results, CloudTrail delivers two types of files to your Amazon S3 bucket.

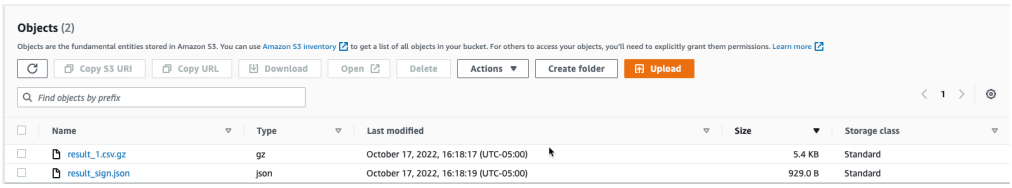
- A sign file in JSON format that you can use to validate the query result files. The sign file is named result_sign.json. For more information about the sign file, see CloudTrail sign file structure (p. 206).
- One or more query result files in CSV format, which contain the results from the query. The number of query result files delivered is dependent upon the total size of the query results. The maximum file size for a query result file is 1 TB. Each query result file is named result_*number*.csv.gz. For example, if the total size of the query results was 2 TB, you would have two query result files, result_1.csv.gz and result_2.csv.gz.

CloudTrail query result and sign files are Amazon S3 objects. You can use the S3 console, the Amazon Command Line Interface (CLI), or the S3 API to retrieve query result and sign files.

The following procedure describes how to download the query result and sign files with the Amazon S3 console.

To download your query result or sign file with the Amazon S3 console

1. Open the Amazon S3 console.
2. Choose the bucket and choose the file that you want to download.



3. Choose **Download** and follow any prompts to save the file.

   Note
   Some browsers, such as Chrome, automatically extract the query result file for you. If your browser does this for you, skip to step 5.
4. Use a product such as 7-Zip to extract the query result file.
5. Open the query result or sign file.

Validate saved query results

To determine whether the query results were modified, deleted, or unchanged after CloudTrail delivered the query results, you can use CloudTrail query results integrity validation. This feature is built using industry standard algorithms: SHA-256 for hashing and SHA-256 with RSA for digital signing. This makes it computationally infeasible to modify, delete or forge CloudTrail query result files without detection. You can use the command line to validate the query result files.

Why use it?

Validated query result files are invaluable in security and forensic investigations. For example, a validated query result file enables you to assert positively that the query result file itself has not changed. The CloudTrail query result file integrity validation process also lets you know if a query result file has been deleted or changed.

Topics

- Validate saved query results with the Amazon CLI (p. 204)
- CloudTrail sign file structure (p. 206)
- Custom implementations of CloudTrail query result file integrity validation (p. 207)

Validate saved query results with the Amazon CLI

You can validate the integrity of the query result files and sign file by using the **aws cloudtrail verify-query-results** command.

Prerequisites

To validate query results integrity with the command line, the following conditions must be met:

- You must have online connectivity to Amazon.
- You must use Amazon CLI version 2.
- To validate query result files and sign file locally, the following conditions apply:
  - You must put the query result files and sign file in the specified file path. Specify the file path as the value for the **--local-export-path** parameter.
  - You must not rename the query result files and sign file.

- To validate the query result files and sign file in the S3 bucket, the following conditions apply:
  - You must not rename the query result files and sign file.
  - You must have read access to the Amazon S3 bucket that contains the query result files and sign file.
  - The specified S3 prefix must contain the query result files and sign file. Specify the S3 prefix as the value for the **--s3-prefix** parameter.

verify-query-results

The **verify-query-results** command verifies the hash value of each query result file by comparing the value with the `fileHashValue` in the sign file, and then validating the `hashSignature` in the sign file.

When you verify query results, you can use either the **--s3-bucket** and **--s3-prefix** command line options to validate the query result files and sign file stored in an S3 bucket, or you can use the **--local-export-path** command line option to perform a local validation of the downloaded query result files and sign file.

> **Note**
> The **verify-query-results** command is Region specific. You must specify the **--region** global option to validate query results for a specific Amazon Web Services Region.

The following are the options for the **verify-query-results** command.

--s3-bucket *<string>*

Specifies the S3 bucket name that stores the query result files and sign file. You cannot use this parameter with **--local-export-path**.

--s3-prefix *<string>*

Specifies the S3 path of the S3 folder that contains the query result files and sign file (for example, `s3/path/`). You cannot use this parameter with **--local-export-path**. You do not need to provide this parameter if the files are located in the root directory of the S3 bucket.

--local-export-path *<string>*

Specifies the local directory that contains the query result files and sign file (for example, `/local/path/to/export/file/`). You cannot use this parameter with **--s3-bucket** or **--s3-prefix**.

Examples

The following example validates query results using the **--s3-bucket** and **--s3-prefix** command line options to specify the S3 bucket name and prefix containing the query result files and sign file.

```
aws cloudtrail verify-query-results --s3-bucket bucket_name --s3-prefix prefix --
region region
```

The following example validates downloaded query results using the **--local-export-path** command line option to specify the local path for the query result files and sign file. For more information about downloading query result files, see Download your CloudTrail Lake saved query results (p. 203).

```
aws cloudtrail verify-query-results --local-export-path local_file_path --region region
```

Validation results

The following table describes the possible validation messages for query result files and sign file.

File Type	Validation Message	Description
Sign file	Successfully validated sign and query result files	The sign file signature is valid. The query result files it references can be checked.
Query result file	ValidationError: "File *file_name* has inconsistent hash value with hash value recorded in sign file, hash value in sign file is *expected_hash*, but get *computed_hash*	Validation failed because the hash value for the query result file did not match the fileHashValue in the sign file.
Sign file	ValidationError: Invalid signature in sign file	Validation for the sign file failed because the signature is not valid.

CloudTrail sign file structure

The sign file contains the name of each query result file that was delivered to your Amazon S3 bucket when you saved the query results, the hash value for each query result file, and the digital signature of the file. The digital signature and hash values are used for validating the integrity of the query result files and of the sign file itself.

Sign file location

The sign file is delivered to an Amazon S3 bucket location that follows this syntax.

```
s3://s3-bucket-name/optional-prefix/AWSLogs/aws-account-ID/CloudTrail-Lake/
Query/year/month/date/query-ID/result_sign.json
```

Sample sign file contents

The following example sign file contains information for CloudTrail Lake query results.

```
{
  "version": "1.0",
  "region": "us-east-1",
  "files": [
    {
      "fileHashValue" : "de85a48b8a363033c891abd723181243620a3af3b6505f0a44db77e147e9c188",
      "fileName" : "result_1.csv.gz"
    }
  ],
  "hashAlgorithm" : "SHA-256",
  "signatureAlgorithm" : "SHA256withRSA",
  "queryCompleteTime": "2022-05-10T22:06:30Z",
  "hashSignature" :
 "7664652aaf1d5a17a12ba50abe6aca77c0ec76264bdf7dce71ac6d1c7781117c2a412e5820bccf473b1361306dff648feae20
  "publicKeyFingerprint" : "67b9fa73676d86966b449dd677850753"
}
```

Sign file field descriptions

The following are descriptions for each field in the sign file:

`version`

The version of the sign file.

`region`

The Region for the Amazon account used for saving the query results.

`files.fileHashValue`

The hexadecimal encoded hash value of the compressed query result file content.

`files.fileName`

The name of the query result file.

`hashAlgorithm`

The hash algorithm used to hash the query result file.

`signatureAlgorithm`

The algorithm used to sign the file.

`queryCompleteTime`

Indicates when CloudTrail delivered the query results to the S3 bucket. You can use this value to find the public key.

`hashSignature`

The hash signature for the file.

`publicKeyFingerprint`

The hexadecimal encoded fingerprint of the public key used to sign the file.

Custom implementations of CloudTrail query result file integrity validation

Because CloudTrail uses industry standard, openly available cryptographic algorithms and hash functions, you can create your own tools to validate the integrity of the CloudTrail query result files. When you save query results to an Amazon S3 bucket, CloudTrail delivers a sign file to your S3 bucket. You can implement your own validation solution to validate the signature and query result files. For more information about the sign file, see CloudTrail sign file structure (p. 206).

This topic describes how the sign file is signed, and then details the steps that you will need to take to implement a solution that validates the sign file and the query result files that the sign file references.

Understanding how CloudTrail sign files are signed

CloudTrail sign files are signed with RSA digital signatures. For each sign file, CloudTrail does the following:

1. Creates a hash list containing the hash value for each query result file.
2. Gets a private key unique to the Region.
3. Passes the SHA-256 hash of the string and the private key to the RSA signing algorithm, which produces a digital signature.
4. Encodes the byte code of the signature into hexadecimal format.
5. Puts the digital signature into the sign file.

Contents of the data signing string

The data signing string consists of the hash value for each query result file separated by a space. The sign file lists the `fileHashValue` for each query result file.

Custom validation implementation steps

When implementing a custom validation solution, you will need to validate the sign file and the query result files that it references.

Validate the sign file

To validate a sign file, you need its signature, the public key whose private key was used to sign it, and a data signing string that you compute.

1. Get the sign file.
2. Verify that the sign file has been retrieved from its original location.
3. Get the hexadecimal-encoded signature of the sign file.
4. Get the hexadecimal-encoded fingerprint of the public key whose private key was used to sign the sign file.
5. Retrieve the public key for the time range corresponding to `queryCompleteTime` in the sign file. For the time range, choose a `StartTime` earlier than the `queryCompleteTime` and an `EndTime` later than the `queryCompleteTime`.
6. From among the public keys retrieved, choose the public key whose fingerprint matches the `publicKeyFingerprint` value in the sign file.
7. Using a hash list containing the hash value for each query result file separated by a space, recreate the data signing string used to verify the sign file signature. The sign file lists the `fileHashValue` for each query result file.

   For example, if your sign file's `files` array contains the following three query result files, your hash list is "aaa bbb ccc".

```
“files": [

   {

        "fileHashValue" : “aaa”,

        "fileName" : "result_1.csv.gz"

   },
   {

        "fileHashValue" : “bbb”,

        "fileName" : "result_2.csv.gz"

   },
   {
```

```
        "fileHashValue" : “ccc”,

        "fileName" : "result_3.csv.gz"

    }
],
```

8. Validate the signature by passing in the SHA-256 hash of the string, the public key, and the signature as parameters to the RSA signature verification algorithm. If the result is true, the sign file is valid.

Validate the query result files

If the sign file is valid, validate the query result files that the sign file references. To validate the integrity of a query result file, compute its SHA-256 hash value on its compressed content and compare the results with the `fileHashValue` for the query result file recorded in the sign file. If the hashes match, the query result file is valid.

The following sections describe the validation process in detail.

A. Get the sign file

The first steps are to get the sign file and get the fingerprint of the public key.

1. Get the sign file from your Amazon S3 bucket for the query results that you want to validate.
2. Next, get the `hashSignature` value from the sign file.
3. In the sign file, get the fingerprint of the public key whose private key was used to sign the file from the `publicKeyFingerprint` field.

B. Retrieve the public key for validating the sign file

To get the public key to validate the sign file, you can use either the Amazon CLI or the CloudTrail API. In both cases, you specify a time range (that is, a start time and end time) for the sign file that you want to validate. Use a time range corresponding to the `queryCompleteTime` in the sign file. One or more public keys may be returned for the time range that you specify. The returned keys may have validity time ranges that overlap.

> **Note**
> Because CloudTrail uses different private/public key pairs per Region, each sign file is signed with a private key unique to its Region. Therefore, when you validate a sign file from a particular Region, you must retrieve its public key from the same Region.

Use the Amazon CLI to retrieve public keys

To retrieve a public key for a sign file by using the Amazon CLI, use the `cloudtrail list-public-keys` command. The command has the following format:

```
aws cloudtrail list-public-keys [--start-time <start-time>] [--end-time <end-time>]
```

The start-time and end-time parameters are UTC timestamps and are optional. If not specified, the current time is used, and the currently active public key or keys are returned.

Sample Response

The response will be a list of JSON objects representing the key (or keys) returned:

Use the CloudTrail API to retrieve public keys

To retrieve a public key for a sign file by using the CloudTrail API, pass in start time and end time values to the `ListPublicKeys` API. The `ListPublicKeys` API returns the public keys whose private keys

were used to sign the file within the specified time range. For each public key, the API also returns the corresponding fingerprint.

`ListPublicKeys`

This section describes the request parameters and response elements for the `ListPublicKeys` API.

Note
The encoding for the binary fields for `ListPublicKeys` is subject to change.

Request Parameters

Name	Description
`StartTime`	Optionally specifies, in UTC, the start of the time range to look up the public key for CloudTrail sign file. If StartTime is not specified, the current time is used, and the current public key is returned. Type: DateTime
`EndTime`	Optionally specifies, in UTC, the end of the time range to look up public keys for CloudTrail sign files. If EndTime is not specified, the current time is used. Type: DateTime

Response Elements

`PublicKeyList`, an array of `PublicKey` objects that contains:

Name	Description
`Value`	The DER encoded public key value in PKCS #1 format. Type: Blob
`ValidityStartTime`	The starting time of validity of the public key. Type: DateTime
`ValidityEndTime`	The ending time of validity of the public key. Type: DateTime
`Fingerprint`	The fingerprint of the public key. The fingerprint can be used to identify the public key that you must use to validate the sign file. Type: String

C. Choose the public key to use for validation

From among the public keys retrieved by `list-public-keys` or `ListPublicKeys`, choose the public key whose fingerprint matches the fingerprint recorded in the `publicKeyFingerprint` field of the sign file. This is the public key that you will use to validate the sign file.

D. Recreate the data signing string

Now that you have the signature of the sign file and the associated public key, you need to calculate the data signing string. After you have calculated the data signing string, you will have the inputs needed to verify the signature.

The data signing string consists of the hash value for each query result file separated by a space. After you recreate this string, you can validate the sign file.

E. Validate the sign file

Pass the recreated data signing string, digital signature, and public key to the RSA signature verification algorithm. If the output is true, the signature of the sign file is verified and the sign file is valid.

F. Validate the query result files

After you have validated the sign file, you can validate the query result files it references. The sign file contains the SHA-256 hashes of the query result files. If one of the query result files was modified after CloudTrail delivered it, the SHA-256 hashes will change, and the signature of the sign file will not match.

Use the following procedure to validate the query result files listed in the sign file's `files` array.

1. Retrieve the original hash of the file from the `files.fileHashValue` field in the sign file.
2. Hash the compressed contents of the query result file with the hashing algorithm specified in `hashAlgorithm`.
3. Compare the hash value that you generated for each query result file with the `files.fileHashValue` in the sign file. If the hashes match, the query result files are valid.

Validating signature and query result files offline

When validating sign and query result files offline, you can generally follow the procedures described in the previous sections. However, you must take into account the following information about public keys.

Public keys

In order to validate offline, the public key that you need for validating query result files in a given time range must first be obtained online (by calling `ListPublicKeys`, for example) and then stored offline. This step must be repeated whenever you want to validate additional files outside the initial time range that you specified.

Sample validation snippet

The following sample snippet provides skeleton code for validating CloudTrail sign and query result files. The skeleton code is online/offline agnostic; that is, it is up to you to decide whether to implement it with or without online connectivity to Amazon. The suggested implementation uses the Java Cryptography Extension (JCE) and Bouncy Castle as a security provider.

The sample snippet shows:

- How to create the data signing string used to validate the sign file signature.
- How to verify the sign file's signature.
- How to calculate the hash value for the query result file and compare it with the `fileHashValue` listed in the sign file to verify the authenticity of the query result file.

```
import org.apache.commons.codec.binary.Hex;
import org.bouncycastle.asn1.pkcs.PKCSObjectIdentifiers;
import org.bouncycastle.asn1.pkcs.RSAPublicKey;
import org.bouncycastle.asn1.x509.AlgorithmIdentifier;
import org.bouncycastle.asn1.x509.SubjectPublicKeyInfo;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.json.JSONArray;
import org.json.JSONObject;
```

```
import java.security.KeyFactory;
import java.security.MessageDigest;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
import java.security.spec.X509EncodedKeySpec;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;


public class SignFileValidationSampleCode {

    public void validateSignFile(String s3Bucket, String s3PrefixPath) throws Exception {
        MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");

        // Load the sign file from S3 (using Amazon S3 Client) or from your local copy
        JSONObject signFile = loadSignFileToMemory(s3Bucket, String.format("%s/%s",
 s3PrefixPath, "result_sign.json"));

        // Using the Bouncy Castle provider as a JCE security provider - http://
www.bouncycastle.org/
        Security.addProvider(new BouncyCastleProvider());

        List<String> hashList = new ArrayList<>();

        JSONArray jsonArray = signFile.getJSONArray("files");

        for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject file = jsonArray.getJSONObject(i);
            String fileS3ObjectKey = String.format("%s/%s", s3PrefixPath,
 file.getString("fileName"));

            // Load the export file from S3 (using Amazon S3 Client) or from your local
 copy
            byte[] exportFileContent = loadCompressedExportFileInMemory(s3Bucket,
 fileS3ObjectKey);
            messageDigest.update(exportFileContent);
            byte[] exportFileHash = messageDigest.digest();
            messageDigest.reset();
            byte[] expectedHash = Hex.decodeHex(file.getString("fileHashValue"));

            boolean signaturesMatch = Arrays.equals(expectedHash, exportFileHash);
            if (!signaturesMatch) {
                System.err.println(String.format("Export file: %s/%s hash doesn't match.
\tExpected: %s Actual: %s",
                        s3Bucket, fileS3ObjectKey,
                        Hex.encodeHexString(expectedHash),
 Hex.encodeHexString(exportFileHash)));
            } else {
                System.out.println(String.format("Export file: %s/%s hash match",
                        s3Bucket, fileS3ObjectKey));
            }

            hashList.add(file.getString("fileHashValue"));
        }
        String hashListString = hashList.stream().collect(Collectors.joining(" "));

        /*
            NOTE:
            To find the right public key to verify the signature, call CloudTrail
 ListPublicKey API to get a list
            of public keys, then match by the publicKeyFingerprint in the sign file. Also,
 the public key bytes
```

```
            returned from ListPublicKey API are DER encoded in PKCS#1 format:

            PublicKeyInfo ::= SEQUENCE {
                algorithm       AlgorithmIdentifier,
                PublicKey       BIT STRING
            }

            AlgorithmIdentifier ::= SEQUENCE {
                algorithm       OBJECT IDENTIFIER,
                parameters      ANY DEFINED BY algorithm OPTIONAL
            }
        */
        byte[] pkcs1PublicKeyBytes = getPublicKey(signFile.getString("queryCompleteTime"),
                signFile.getString("publicKeyFingerprint"));
        byte[] signatureContent = Hex.decodeHex(signFile.getString("hashSignature"));

        // Transform the PKCS#1 formatted public key to x.509 format.
        RSAPublicKey rsaPublicKey = RSAPublicKey.getInstance(pkcs1PublicKeyBytes);
        AlgorithmIdentifier rsaEncryption = new
 AlgorithmIdentifier(PKCSObjectIdentifiers.rsaEncryption, null);
        SubjectPublicKeyInfo publicKeyInfo = new SubjectPublicKeyInfo(rsaEncryption,
 rsaPublicKey);

        // Create the PublicKey object needed for the signature validation
        PublicKey publicKey = KeyFactory.getInstance("RSA", "BC")
                .generatePublic(new X509EncodedKeySpec(publicKeyInfo.getEncoded()));

        // Verify signature
        Signature signature = Signature.getInstance("SHA256withRSA", "BC");
        signature.initVerify(publicKey);
        signature.update(hashListString.getBytes("UTF-8"));

        if (signature.verify(signatureContent)) {
            System.out.println("Sign file signature is valid.");
        } else {
            System.err.println("Sign file signature failed validation.");
        }

        System.out.println("Sign file validation completed.");
    }
}
```

Learning resources

The following resources can help you get a better understanding of what CloudTrail Lake is and how you can use it.

- Modernize Your Audit Log Management Using CloudTrail Lake (YouTube video)
- Log Activity Events from Non-Amazon Sources in Amazon CloudTrail Lake (YouTube video)
- Get visibility into the activity logs for your workforce and customer identities (Amazon blog)
- Using Amazon CloudTrail Lake to identify older TLS connections to Amazon service endpoints (Amazon blog)
- How Arctic Wolf uses Amazon CloudTrail Lake to Simplify Security and Operations (Amazon blog)
- CloudTrail Lake FAQs
- Amazon CloudTrail API Reference
- Amazon CloudTrail Data API Reference
- Amazon CloudTrail Partner Onboarding Guide

Managing CloudTrail Lake by using the Amazon CLI

The following are example Amazon CLI commands for creating and managing event data stores and queries in CloudTrail Lake.

Topics

- Create an event data store for CloudTrail data events (p. 214)
- Create an event data store for Amazon Config configuration items (p. 215)
- Create an integration to log events from outside Amazon (p. 216)
- Get an event data store (p. 221)
- List all event data stores in an account (p. 222)
- Update an event data store (p. 222)
- Stop ingestion on an event data store (p. 223)
- Start ingestion on an event data store (p. 223)
- Delete an event data store (p. 224)
- Restore an event data store (p. 224)
- List all channels (p. 224)
- Update a channel (p. 224)
- Delete a channel to delete an integration (p. 225)
- Start a query (p. 225)
- Get metadata about a query (p. 225)
- Get query results (p. 226)
- List all queries on an event data store (p. 226)
- Cancel a running query (p. 227)

Create an event data store for CloudTrail data events

The following example Amazon Command Line Interface (Amazon CLI) command creates an event data store named `my-event-data-store` that selects all Amazon S3 data events. The event data store retention period in this example is 90 days.`--name` is required; other parameters are optional. Valid values for `--retention-period` are integers between 7 and 2557, representing days. If you do not specify `--retention-period`, CloudTrail uses the default retention period of 2557 days. By default, `--multi-region-enabled` is set, even if the parameter is not added, and the event data store includes events from all Regions. The event data store is not enabled for all accounts in an Amazon Organizations organization by default. To enable an event data store to collect events for all accounts in an organization, add the `--organization-enabled` parameter. The parameters `--termination-protection-enabled` (the default) and `--no-termination-protection-enabled` set and remove termination protection, respectively. Optionally, you can choose to enable Amazon Key Management Service encryption and specify an Amazon KMS key by adding `--kms-key-id` to the command, and specifying a KMS key ARN as the value. The `--advanced-event-selectors` parameter includes or excludes data events in your event data store. For more information about `--advanced-event-selectors`, see `AdvancedEventSelectors` in the CloudTrail API Reference.

```
aws cloudtrail create-event-data-store \
--name my-event-data-store \
--retention-period 90 \
--kms-key-id "arn:aws:kms:us-east-1:123456789012:alias/KMS_key_alias" \
--advanced-event-selectors '[
        {
            "Name": "Select all S3 data events",
            "FieldSelectors": [
```

```
                { "Field": "eventCategory", "Equals": ["Data"] },
                { "Field": "resources.type", "Equals": ["AWS::S3::Object"] },
                { "Field": "resources.ARN", "StartsWith": ["arn:aws:s3"] }
            ]
        }
    ]'
```

The following is an example response.

```
{
    "EventDataStoreArn": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
ee54-4813-92d5-999aeEXAMPLE",
    "Name": "my-event-data-store",
    "Status": "CREATED",
    "AdvancedEventSelectors": [
        {
            "Name": "Select all S3 data events",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "StartsWith": [
                        "arn:aws:s3"
                    ]
                }
            ]
        }
    ],
    "MultiRegionEnabled": true,
    "OrganizationEnabled": false,
    "RetentionPeriod": 90,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:alias/KMS_key_alias",
    "TerminationProtectionEnabled": true,
    "CreatedTimestamp": "2021-12-09T22:19:39.417000-05:00",
    "UpdatedTimestamp": "2021-12-09T22:19:39.603000-05:00"
}
```

Create an event data store for Amazon Config configuration items

The following example Amazon CLI **create-event-data-store** command creates an event data store named `my-event-data-store` that selects Amazon Config configuration items. The event data store retention period in this example is 90 days. The `--name` parameter is required; other parameters are optional. Valid values for `--retention-period` are integers between 7 and 2557, representing days. If you don't specify `--retention-period`, CloudTrail uses the default retention period of 2557 days. By default, `--multi-region-enabled` is set, even if the parameter is not added, and the event data store includes configuration items from all Amazon Web Services Regions. The event data store is not enabled for all accounts in an Amazon Organizations organization by default; to enable an event data store to collect configuration items for all accounts in an organization, add the `--organization-`

enabled parameter. The parameters --termination-protection-enabled (the default) and --no-termination-protection-enabled set and remove termination protection, respectively. Optionally, you can choose to enable Amazon Key Management Service encryption and specify an Amazon KMS key by adding --kms-key-id to the command, and specifying a KMS key ARN as the value. The --advanced-event-selectors parameter includes configuration items in your event data store by specifying the eventCategory field equals ConfigurationItem.

```
aws cloudtrail create-event-data-store
--name my-event-data-store
--retention-period 90
--kms-key-id "arn:aws:kms:us-east-1:123456789012:alias/KMS_key_alias" \
--advanced-event-selectors '[
        {
            "Name": "Select Amazon Config configuration items",
            "FieldSelectors": [
                { "Field": "eventCategory", "Equals": ["ConfigurationItem"] }
            ]
        }
    ]'
```

The following is an example response.

```
{
    "EventDataStoreArn": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
ee54-4813-92d5-999aeEXAMPLE",
    "Name": "my-event-data-store",
    "Status": "CREATED",
    "AdvancedEventSelectors": [
        {
            "Name": "Select Amazon Config configuration items",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "ConfigurationItem"
                    ]
                }
            ]
        }
    ],
    "MultiRegionEnabled": true,
    "OrganizationEnabled": false,
    "RetentionPeriod": 90,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:alias/KMS_key_alias",
    "TerminationProtectionEnabled": true,
    "CreatedTimestamp": "2022-10-07T19:03:24.277000+00:00",
    "UpdatedTimestamp": "2022-10-07T19:03:24.468000+00:00"
}
```

Create an integration to log events from outside Amazon

In the Amazon CLI, you create an integration that logs events from outside Amazon in four commands (three if you already have an event data store that meets the criteria). Event data stores that you use as the destinations for an integration must be for a single Region and single account; they cannot be multi-region, they cannot log events for organizations in Amazon Organizations, and they can only include activity events. The event type in the console must be **Events from integrations**. In the API, the eventCategory value must be ActivityAuditLog. For more information about integrations, see Create an integration with an event source outside of Amazon (p. 180).

1. Run **create-event-data-store** to create an event data store, if you do not already have one or more event data stores that you can use for the integration.

   The following example Amazon CLI command creates an event data store that logs events from outside Amazon. For activity events, the `eventCategory` field selector value is `ActivityAuditLog`. The event data store has a retention period of 90 days set. By default, the event data store collects events from all Regions, but because this is collecting non-Amazon events, set it to a single Region by adding the `--no-multi-region-enabled` option. Termination protection is enabled by default, and the event data store does not collect events for accounts in an organization.

```
aws cloudtrail create-event-data-store \
--name my-event-data-store \
--no-multi-region-enabled \
--retention-period 90 \
--advanced-event-selectors '[
    {
      "Name": "Select all external events",
      "FieldSelectors": [
          { "Field": "eventCategory", "Equals": ["ActivityAuditLog"] }
        ]
    }
  ]'
```

   The following is an example response.

```
{
    "Name": "my-event-data-store",
    "ARN": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/
EXAMPLEf852-4e8f-8bd1-bcf6cEXAMPLE",
    "RetentionPeriod": "90",
    "MultiRegionEnabled": false,
    "OrganizationEnabled": false,
    "TerminationProtectionEnabled": true,
    "AdvancedEventSelectors": [
        {
           "Name": "Select all external events",
           "FieldSelectors": [
              {
                  "Field": "eventCategory",
                  "Equals": [
                      "ActivityAuditLog"
                    ]
                }
             ]
        }
    ]
}
```

   You'll need the event data store ID (the suffix of the ARN, or `EXAMPLEf852-4e8f-8bd1-bcf6cEXAMPLE` in the preceding response example) to go on to the next step and create your channel.

2. Run the **create-channel** command to create a channel that allows a partner or source application to send events to an event data store in CloudTrail.

   A channel has the following components:

Source

CloudTrail uses this information to determine the partners that are sending event data to CloudTrail on your behalf. A source is required, and can be either `Custom` for all valid non-Amazon events, or the name of a partner event source. A maximum of one channel is allowed per source.

For information about the `Source` values for available partners, see Additional information about integration partners (p. 185).

Ingestion status

The channel status shows when the last events were received from a channel source.

Destinations

The destinations are the CloudTrail Lake event data stores that are receiving events from the channel. You can change destination event data stores for a channel.

To stop receiving events from a source, delete the channel.

You need the ID of at least one destination event data store to run this command. The valid type of destination is `EVENT_DATA_STORE`. You can send ingested events to more than one event data store. The following example command creates a channel that sends events to two event data stores, represented by their IDs in the `Location` attribute of the `--destinations` parameter. The `--destinations`, `--name`, and `--source` parameters are required. To ingest events from a CloudTrail partner, specify the name of the partner as the value of `--source`. To ingest events from your own applications outside Amazon, specify `Custom` as the value of `--source`.

```
aws cloudtrail create-channel \
    --region us-east-1 \
    --destinations '[{"Type": "EVENT_DATA_STORE", "Location": "EXAMPLEf852-4e8f-8bd1-
bcf6cEXAMPLE"}, {"Type": "EVENT_DATA_STORE", "Location": "EXAMPLEg922-5n2l-3vz1-
 apqw8EXAMPLE"}]'
    --name my-partner-channel \
    --source $partnerSourceName \
```

In the response to your **create-channel** command, copy the ARN of the new channel. You need the ARN to run the `put-resource-policy` and `put-audit-events` commands in the next steps.

3. Run the **put-resource-policy** command to attach a resource policy to the channel. Resource policies are JSON policy documents that specify what actions a specified principal can perform on the resource and under what conditions. The accounts defined as principals in the channel's resource policy can call the `PutAuditEvents` API to deliver events.

   Note
   If you do not create a resource policy for the channel, only the channel owner can call the `PutAuditEvents` API on the channel.

   The information required for the policy is determined by the integration type.

   - For a direction integration, CloudTrail requires the policy to contain the partner's Amazon account IDs, and requires you to enter the unique external ID provided by the partner. CloudTrail automatically adds the partner's Amazon account IDs to the resource policy when you create an integration using the CloudTrail console. Refer to the partner's documentation to learn how to get the Amazon account numbers required for the policy.
   - For a solution integration, you must specify at least one Amazon account ID as principal, and can optionally enter an external ID to prevent against confused deputy.

The following are requirements for the resource policy:

- The resource ARN defined in the policy must match the channel ARN the policy is attached to.
- The policy contains only one action: cloudtrail-data:PutAuditEvents
- The policy contains at least one statement. The policy can have a maximum of 20 statements.
- Each statement contains at least one principal. A statement can have a maximum of 50 principals.

```
aws cloudtrail put-resource-policy \
    --resource-arn "channelARN" \
    --policy "{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Sid": "ChannelPolicy",
            "Effect": "Allow",
            "Principal":
            {
                "AWS":
                [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root",
                    "arn:aws:iam::123456789012:root"
                ]
            },
            "Action": "cloudtrail-data:PutAuditEvents",
            "Resource": "arn:aws:cloudtrail:us-east-1:777788889999:channel/
EXAMPLE-80b5-40a7-ae65-6e099392355b",
            "Condition":
            {
                "StringEquals":
                {
                    "cloudtrail:ExternalId": "UniqueExternalIDFromPartner"
                }
            }
        }
    ]
}"
```

For more information about resource policies, see Amazon CloudTrail resource-based policy examples (p. 361).

4. Run the `PutAuditEvents` API to ingest your activity events into CloudTrail. You'll need the payload of events that you want CloudTrail to add. Be sure that there is no sensitive or personally-identifying information in event payload before ingesting it into CloudTrail. Note that the `PutAuditEvents` API uses the `cloudtrail-data` CLI endpoint, not the `cloudtrail` endpoint.

   The following examples show how to use the **put-audit-events** CLI command. The **--audit-events** and **--channel-arn** parameters are required. The **--external-id** parameter is required if an external ID is defined in the resource policy. You need the ARN of the channel that you created in the preceding step. The value of **--audit-events** is a JSON array of event objects. `--audit-events` includes a required ID from the event, the required payload of the event as the value of `EventData`, and an optional checksum (p.) to help validate the integrity of the event after ingestion into CloudTrail.

```
aws cloudtrail-data put-audit-events \
--channel-arn $ChannelArn \
```

```
--external-id $UniqueExternalIDFromPartner \
--audit-events \
id="event_ID",eventData='"{event_payload}"' \
id="event_ID",eventData='"{event_payload}"',eventDataChecksum="optional_checksum"
```

The following is an example command with two event examples.

```
aws cloudtrail-data put-audit-events \
--channel-arn arn:aws:cloudtrail:us-east-1:123456789012:channel/EXAMPLE8-0558-4f7e-
a06a-43969EXAMPLE \
--external-id UniqueExternalIDFromPartner \
--audit-events \
id="EXAMPLE3-0f1f-4a85-9664-d50a3EXAMPLE",eventData='"{\"eventVersion\":\0.01\",
\"eventSource\":\"custom1.domain.com\", ...
\}"' \
id="EXAMPLE7-a999-486d-b241-b33a1EXAMPLE",eventData='"{\"eventVersion\":\0.02\",
\"eventSource\":\"custom2.domain.com\", ...
\}"',eventDataChecksum="EXAMPLE6e7dd61f3ead...93a691d8EXAMPLE"
```

The following example command adds the `--cli-input-json` parameter to specify a JSON file (`custom-events.json`) of event payload.

```
aws cloudtrail-data put-audit-events --channel-arn $channelArn --external-id
 $UniqueExternalIDFromPartner --cli-input-json file://custom-events.json --region us-
east-1
```

The following are the sample contents of the example JSON file, `custom-events.json`.

```
{
    "auditEvents": [
      {
        "eventData": "{\"version\":\"eventData.version\",\"UID\":\"UID\",
        \"userIdentity\":{\"type\":\"CustomUserIdentity\",\"principalId\":\"principalId
\",
        \"details\":{\"key\":\"value\"}},\"eventTime\":\"2021-10-27T12:13:14Z\",
\"eventName\":\"eventName\",
        \"userAgent\":\"userAgent\",\"eventSource\":\"eventSource\",
        \"requestParameters\":{\"key\":\"value\"},\"responseElements\":{\"key\":\"value
\"},
        \"additionalEventData\":{\"key\":\"value\"},
        \"sourceIPAddress\":\"12.34.56.78\",\"recipientAccountId\":\"152089810396\"}",
        "id": "1"
      }
   ]
}
```

You can verify that the integration is working, and CloudTrail is ingesting events from the source correctly, by running the **get-channel** command. The output of **get-channel** shows the most recent time stamp that CloudTrail received events.

```
aws cloudtrail get-channel --channel arn:aws:cloudtrail:us-east-1:01234567890:channel/
EXAMPLE8-0558-4f7e-a06a-43969EXAMPLE
```

(Optional) Calculate a checksum value

The checksum that you specify as the value of `EventDataChecksum` in a `PutAuditEvents` request helps you verify that CloudTrail receives the event that matches with the checksum; it helps verify the

integrity of events. The checksum value is a base64-SHA256 algorithm that you calculate by running the following command.

```
printf %s "{"eventData": "{\"version\":\"eventData.version\",\"UID\":\"UID\",
        \"userIdentity\":{\"type\":\"CustomUserIdentity\",\"principalId\":\"principalId\",
        \"details\":{\"key\":\"value\"}},\"eventTime\":\"2021-10-27T12:13:14Z\",\"eventName
\":\"eventName\",
        \"userAgent\":\"userAgent\",\"eventSource\":\"eventSource\",
        \"requestParameters\":{\"key\":\"value\"},\"responseElements\":{\"key\":\"value\"},
        \"additionalEventData\":{\"key\":\"value\"},
        \"sourceIPAddress\":\"source_IP_address\",
        \"recipientAccountId\":\"recipient_account_ID\"}",
        "id": "1"}" \
 | openssl dgst -binary -sha256 | base64
```

The command returns the checksum. The following is an example.

```
EXAMPLEDHjkI8iehvCUCWTIAbNYkOgO/t0YNw+7rrQE=
```

The checksum value becomes the value of `EventDataChecksum` in your `PutAuditEvents` request. If the checksum doesn't match with the one for the provided event, CloudTrail rejects the event with an `InvalidChecksum` error.

Get an event data store

The following example Amazon CLI **get-event-data-store** command returns information about the event data store specified by the required `--event-data-store` parameter, which accepts an ARN or the ID suffix of the ARN.

```
aws cloudtrail get-event-data-store
--event-data-store arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE
```

The following is an example response. Creation and last updated times are in `timestamp` format.

```
{
    "EventDataStoreARN": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE",
    "Name": "my-event-data-store",
    "Status": "Enabled",
    "KmsKeyId": "kms_key_ID",
    "AdvancedEventSelectors": [
        {
            "Name": "Select All S3 Data Events",
            "FieldSelectors": [
                { "Field": "eventCategory", "Equals": ["Data"] },
                { "Field": "resources.type", "Equals": ["AWS::S3::Object"] },
                { "Field": "resources.ARN", "StartsWith": ["arn:aws:s3"] }
            ]
        }
    ]
    "CreatedTimestamp": 1248496624,
    "UpdatedTimestamp": 1598296624,
    "MultiRegionEnabled": true,
    "RetentionPeriod": 90,
    "TerminationProtectionEnabled": true,
}
```

List all event data stores in an account

The following example Amazon CLI **list-event-data-stores** command returns information about all event data stores in an account, in the current Region. Optional parameters include `--max-results`, to specify a maximum number of results that you want the command to return on a single page. If there are more results than your specified `--max-results` value, run the command again adding the returned `NextToken` value to get the next page of results.

```
aws cloudtrail list-event-data-stores
```

The following is an example response.

```
{
    "EventDataStores": [
        {
            "EventDataStoreArn": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/
EXAMPLE-ee54-4813-92d5-999aeEXAMPLE",
            "Name": "my-event-data-store"
        }
    ]
}
```

Update an event data store

The following example Amazon CLI **update-event-data-store** command updates an event data store to change its retention period to 100 days, and enable termination protection. The required `--event-data-store` parameter value is an ARN (or the ID suffix of the ARN) and is required; other parameters are optional. In this example, the `--retention-period` parameter is added to change the retention period to 100 days. Optionally, you can choose to enable Amazon Key Management Service encryption and specify an Amazon KMS key by adding `--kms-key-id` to the command, and specifying a KMS key ARN as the value. `--termination-protection-enabled` is added to enable termination protection on an event data store that did not have termination protection enabled.

An event data store that logs events from outside Amazon cannot be updated to log Amazon events. Similarly, an event data store that logs Amazon events cannot be updated to log events from outside Amazon.Amazon

> **Note**
> If you decrease the retention period of an event data store, CloudTrail will remove any events with an `eventTime` older than the new retention period. For example, if the previous retention period was 365 days and you decrease it to 100 days, CloudTrail will remove events with an `eventTime` older than 100 days.

```
aws cloudtrail update-event-data-store
--event-data-store arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE
--retention-period 100
--kms-key-id "arn:aws:kms:us-east-1:0123456789:alias/KMS_key_alias" \
--termination-protection-enabled
```

The following is an example response.

```
{
    "EventDataStoreArn": "arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
ee54-4813-92d5-999aeEXAMPLE",
    "Name": "my-event-data-store",
    "Status": "ENABLED",
```

```
    "KmsKeyId": "kms_key_ID",
    "AdvancedEventSelectors": [
        {
            "Name": "Select all S3 data events",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "StartsWith": [
                        "arn:aws:s3"
                    ]
                }
            ]
        }
    ],
    "MultiRegionEnabled": true,
    "OrganizationEnabled": false,
    "RetentionPeriod": 100,
    "KmsKeyId": "arn:aws:kms:us-east-1:0123456789:alias/KMS_key_alias"
    "TerminationProtectionEnabled": true,
    "CreatedTimestamp": "2021-12-09T22:19:39.417000-05:00",
    "UpdatedTimestamp": "2021-12-09T22:19:39.603000-05:00"
}
```

Stop ingestion on an event data store

The following example Amazon CLI **stop-event-data-store-ingestion** command stops an event data store from ingesting events. To stop ingestion, the event data store `Status` must be `ENABLED` and the `eventCategory` must be `Management`, `Data`, or `ConfigurationItem`. The event data store is specified by `--event-data-store`, which accepts an event data store ARN, or the ID suffix of the ARN. After you run **stop-event-data-store-ingestion**, the state of the event data store changes to `STOPPED_INGESTION`.

The event data store does count towards your account maximum of ten event data stores when its state is `STOPPED_INGESTION`.

```
aws cloudtrail stop-event-data-store-ingestion
--event-data-store arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE
```

There is no response if the operation is successful.

Start ingestion on an event data store

The following example Amazon CLI **start-event-data-store-ingestion** command starts event ingestion on an event data store. To start ingestion, the event data store `Status` must be `STOPPED_INGESTION` and the `eventCategory` must be `Management`, `Data`, or `ConfigurationItem`. The event data store is specified by `--event-data-store`, which accepts an event data store ARN, or the ID suffix of the ARN. After you run **start-event-data-store-ingestion**, the state of the event data store changes to `ENABLED`.

```
aws cloudtrail start-event-data-store-ingestion
--event-data-store arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE
```

There is no response if the operation is successful.

Delete an event data store

The following example Amazon CLI **delete-event-data-store** command disables the event data store specified by `--event-data-store`, which accepts an event data store ARN, or the ID suffix of the ARN. After you run **delete-event-data-store**, the final state of the event data store is `PENDING_DELETION`, and the event data store is automatically deleted after a wait period of seven days. `--no-termination-protection-enabled` must be set on the event data store; this operation cannot work if `--termination-protection-enabled` is set.

After you run **delete-event-data-store** on an event data store, you cannot run **list-queries**, **describe-query**, or **get-query-results** on queries that are using the disabled data store. The event data store does count towards your account maximum of ten event data stores when it is pending deletion.

```
aws cloudtrail delete-event-data-store
--event-data-store arn:aws:cloudtrail:us-east-1:123456789012:eventdatastore/EXAMPLE-
f852-4e8f-8bd1-bcf6cEXAMPLE
```

There is no response if the operation is successful.

Restore an event data store

The following example Amazon CLI **restore-event-data-store** command restores an event data store that is pending deletion. The event data store is specified by `--event-data-store`, which accepts an event data store ARN or the ID suffix of the ARN. You can only restore a deleted event data store within the seven-day wait period after deletion.

```
aws cloudtrail restore-event-data-store
--event-data-store EXAMPLE-f852-4e8f-8bd1-bcf6cEXAMPLE
```

The response includes information about the event data store, including its ARN, advanced event selectors, and the status of restoration.

List all channels

To list all channels in your account, run the **list-channels** command. The following is an example.

```
aws cloudtrail list-channels
```

Update a channel

To update a channel's name or destination event data stores, run the **update-channel** command. The `--channel` parameter is required. You cannot update the source of a channel. The following is an example.

```
aws cloudtrail update-channel \
--channel aws:cloudtrail:us-east-1:123456789012:channel/EXAMPLE8-0558-4f7e-
a06a-43969EXAMPLE \
--name "new-channel-name" \
```

```
--destinations '[{"Type": "EVENT_DATA_STORE", "Location": "EXAMPLEf852-4e8f-8bd1-
bcf6cEXAMPLE"}, {"Type": "EVENT_DATA_STORE", "Location": "EXAMPLEg922-5n21-3vz1-
 apqw8EXAMPLE"}]'
```

Delete a channel to delete an integration

To stop ingesting partner or other activity events outside Amazon, delete the channel by running the **delete-channel** command. The ARN or channel ID (the ARN suffix) of the channel that you want to delete is required. The following is an example.

```
aws cloudtrail delete-channel \
--channel EXAMPLE8-0558-4f7e-a06a-43969EXAMPLE
```

Start a query

The following example Amazon CLI **start-query** command runs a query on the event data store specified as an ID in the query statement and delivers the query results to a specified S3 bucket. The `--query-statement` parameter provides a SQL query, enclosed in single quotation marks. Optional parameters include `--delivery-s3uri`, to deliver the query results to a specified S3 bucket. For more information about the query language you can use in CloudTrail Lake, see CloudTrail Lake SQL constraints (p. 227).

```
aws cloudtrail start-query
--query-statement 'SELECT eventID, eventTime FROM EXAMPLE-f852-4e8f-8bd1-bcf6cEXAMPLE LIMIT
 10'
--delivery-s3uri "s3://aws-cloudtrail-lake-query-results-123456789012-us-east-1"
```

The response is a `QueryId` string. To get the status of a query, run **describe-query** using the `QueryId` value returned by **start-query**. If the query is successful, you can run **get-query-results** to get results.

Output

```
{
    "QueryId": "EXAMPLE2-0add-4207-8135-2d8a4EXAMPLE"
}
```

> **Note**
> Queries that run for longer than one hour might time out. You can still get partial results that were processed before the query timed out.
> If you are delivering the query results to an S3 bucket using the optional `--delivery-s3uri` parameter, the bucket policy must grant CloudTrail permission to delivery query results to the bucket. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail Lake query results (p. 368).

Get metadata about a query

The following example Amazon CLI **describe-query** command gets metadata about a query, including query run time in milliseconds, number of events scanned and matched, total number of bytes scanned, and query status. The `BytesScanned` value matches the number of bytes for which your account is billed for the query, unless the query is still running. If the query results were delivered to an S3 bucket, the response also provides the S3 URI and the delivery status.

You must specify a value for either the `--query-id` or the `--query-alias` parameter. Specifying the `--query-alias` parameter returns information about the last query run for the alias.

```
aws cloudtrail describe-query
```

```
--query-id EXAMPLEd-17a7-47c3-a9a1-eccf7EXAMPLE
```

The following is an example response.

```
{
    "QueryId": "EXAMPLE2-0add-4207-8135-2d8a4EXAMPLE",
    "QueryString": "SELECT eventID, eventTime FROM EXAMPLE-f852-4e8f-8bd1-bcf6cEXAMPLE
 LIMIT 10",
    "QueryStatus": "RUNNING",
    "QueryStatistics": {
        "EventsMatched": 10,
        "EventsScanned": 1000,
        "BytesScanned": 35059,
        "ExecutionTimeInMillis": 3821,
        "CreationTime": "1598911142"
    }
}
```

Get query results

The following example Amazon CLI **get-query-results** command gets event data results of a query. You must specify the `--query-id` returned by the **start-query** command. The `BytesScanned` value matches the number of bytes for which your account is billed for the query, unless the query is still running. Optional parameters include `--max-query-results`, to specify a maximum number of results that you want the command to return on a single page. If there are more results than your specified `--max-query-results` value, run the command again adding the returned `NextToken` value to get the next page of results.

```
aws cloudtrail get-query-results
--query-id EXAMPLEd-17a7-47c3-a9a1-eccf7EXAMPLE
```

Output

```
{
    "QueryStatus": "RUNNING",
    "QueryStatistics": {
        "ResultsCount": 244,
        "TotalResultsCount": 1582,
        "BytesScanned":27044
    },
    "QueryResults": [
      {
        "key": "eventName",
        "value": "StartQuery",
      }
   ],
    "QueryId": "EXAMPLE2-0add-4207-8135-2d8a4EXAMPLE",
    "QueryString": "SELECT eventID, eventTime FROM EXAMPLE-f852-4e8f-8bd1-bcf6cEXAMPLE
 LIMIT 10",
    "NextToken": "20add42078135EXAMPLE"
}
```

List all queries on an event data store

The following example Amazon CLI **list-queries** command returns a list of queries and query statuses on a specified event data store for the past seven days. You must specify an ARN or the ID suffix of an ARN value for `--event-data-store`. Optionally, to shorten the list of results, you can specify a time range, formatted as timestamps, by adding `--start-time` and `--end-time` parameters, and a `--`

`query-status` value. Valid values for `QueryStatus` include `QUEUED`, `RUNNING`, `FINISHED`, `FAILED`, or `CANCELLED`.

list-queries also has optional pagination parameters. Use `--max-results` to specify a maximum number of results that you want the command to return on a single page. If there are more results than your specified `--max-results` value, run the command again adding the returned `NextToken` value to get the next page of results.

```
aws cloudtrail list-queries
--event-data-store EXAMPLE-f852-4e8f-8bd1-bcf6cEXAMPLE
--query-status CANCELLED
--start-time 1598384589
--end-time 1598384602
--max-results 10
```

Output

```
{
    "Queries": [
        {
          "QueryId": "EXAMPLE2-0add-4207-8135-2d8a4EXAMPLE",
          "QueryStatus": "CANCELLED",
          "CreationTime": 1598911142
        },
        {
          "QueryId": "EXAMPLE2-4e89-9230-2127-5dr3aEXAMPLE",
          "QueryStatus": "CANCELLED",
          "CreationTime": 1598296624"
        }
     ],
    "NextToken": "20add42078135EXAMPLE"
}
```

Cancel a running query

The following example Amazon CLI **cancel-query** command cancels a query with a status of `RUNNING`. You must specify a value for `--query-id`. When you run **cancel-query**, the query status might show as `CANCELLED` even if the **cancel-query** operation is not yet finished.

> **Note**
> A canceled query can incur charges. Your account is still charged for the amount of data that was scanned before you canceled the query.

The following is a CLI example.

```
aws cloudtrail cancel-query
--query-id EXAMPLEd-17a7-47c3-a9a1-eccf7EXAMPLE
```

Output

```
QueryId -> (string)
QueryStatus -> (string)
```

CloudTrail Lake SQL constraints

CloudTrail Lake queries are SQL strings. This section provides information about the supported functions, operators, and schemas.

Only SELECT statements are allowed. No query strings can change or mutate data.

CloudTrail Lake supports all valid Presto SQL SELECT statements, functions, and operators. For more information about the supported SQL functions and operators, see Functions and Operators on the Presto 0.280 documentation website.

The CloudTrail console provides a number of sample queries that can help you get started writing your own queries. For more information, see Viewing sample queries in the CloudTrail console (p. 196).

Topics

- Supported functions, condition and join operators (p. 228)
- Advanced, multi-table query support (p. 229)
- Supported schemas for event data stores (p. 229)

Supported functions, condition and join operators

Supported functions

CloudTrail Lake supports all Presto functions. For more information about the supported functions, see Functions and Operators on the Presto 0.280 documentation website.

CloudTrail Lake does not support the INTERVAL keyword.

Supported condition operators

The following are supported condition operators.

```
AND
OR
IN
NOT
IS (NOT) NULL
LIKE
BETWEEN
GREATEST
LEAST
IS DISTINCT FROM
IS NOT DISTINCT FROM
<
>
<=
>=
<>
!=
( conditions ) #parenthesised conditions
```

Supported join operators

The following are the supported JOIN operators. For more information about running multi-table queries, see Advanced, multi-table query support (p. 229).

```
UNION
UNION ALL
EXCEPT
INTERSECT
LEFT JOIN
RIGHT JOIN
INNER JOIN
```

Advanced, multi-table query support

CloudTrail Lake supports advanced query language across multiple event data stores. Only queries that do not include sub-queries are supported.

- UNION|UNION ALL|EXCEPT|INTERSECT (p. 229)
- LEFT|RIGHT|INNER JOIN (p. 229)

To run your query, use the **start-query** command in the Amazon CLI. The following is an example, using one of the sample queries in this section.

```
aws cloudtrail start-query
--query-statement "Select eventId, eventName from EXAMPLEf852-4e8f-8bd1-bcf6cEXAMPLE
 UNION Select eventId, eventName from EXAMPLEg741-6y1x-9p3v-bnh6iEXAMPLE UNION ALL Select
 eventId, eventName from EXAMPLEb529-4e8f9l3d-6m2z-1kp5sEXAMPLE ORDER BY eventId LIMIT 10;"
```

The response is a `QueryId` string. To get the status of a query, run `describe-query`, using the `QueryId` value returned by `start-query`. If the query is successful, you can run `get-query-results` to get results.

UNION|UNION ALL|EXCEPT|INTERSECT

This release adds *multi-table* queries, or queries that you can run across multiple event data stores. The following is an example query that uses `UNION` and `UNION ALL` to find events by their event ID and event name in three event data stores, EDS1, EDS2, and EDS3. The results are selected from each event data store first, then results are concatenated, ordered by event ID, and limited to ten events.

```
Select eventId, eventName from EDS1
UNION
Select eventId, eventName from EDS2
UNION ALL
Select eventId, eventName from EDS3
ORDER BY eventId LIMIT 10;
```

LEFT|RIGHT|INNER JOIN

This release adds *multi-table* queries, or queries that you can run across multiple event data stores. The following is an example query that uses `LEFT JOIN` to find all events from an event data store named `eds2`, mapped to `edsB`, that match those in a primary (left) event data store, `edsA`. The returned events occur on or before January 1, 2020, and only the event names are returned.

```
SELECT edsA.eventName, edsB.eventName, element_at(edsA.map, 'test')
FROM eds1 as edsA
LEFT JOIN eds2 as edsB
ON edsA.eventId = edsB.eventId
WHERE edsA.eventtime <= '2020-01-01'
ORDER BY edsB.eventName;
```

Supported schemas for event data stores

The following sections provide the supported SQL schema for each event data store type.

Topics

- Supported schema for CloudTrail event record fields (p. 230)
- Supported schema for Amazon Config configuration item record fields (p. 232)
- Supported schema for Amazon Audit Manager evidence record fields (p. 233)
- Supported schema for non-Amazon event fields (p. 234)

Supported schema for CloudTrail event record fields

The following is the valid SQL schema for CloudTrail event record fields. For more information about CloudTrail event record fields, see CloudTrail record contents (p. 402).

```
[
    {
        "Name": "eventversion",
        "Type": "string"
    },
    {
        "Name": "useridentity",
        "Type":
 "struct<type:string,principalid:string,arn:string,accountid:string,accesskeyid:string,

 username:string,sessioncontext:struct<attributes:struct<creationdate:timestamp,

 mfaauthenticated:string>,sessionissuer:struct<type:string,principalid:string,arn:string,

 accountid:string,username:string>,webidfederationdata:struct<federatedprovider:string,

 attributes:map<string,string>>,sourceidentity:string,ec2roledelivery:string,
                ec2issuedinvpc:string>,invokedby:string,identityprovider:string>"
    },
    {
        "Name": "eventtime",
        "Type": "timestamp"
    },
    {
        "Name": "eventsource",
        "Type": "string"
    },
    {
        "Name": "eventname",
        "Type": "string"
    },
    {
        "Name": "awsregion",
        "Type": "string"
    },
    {
        "Name": "sourceipaddress",
        "Type": "string"
    },
    {
        "Name": "useragent",
        "Type": "string"
    },
    {
        "Name": "errorcode",
        "Type": "string"
    },
    {
        "Name": "errormessage",
        "Type": "string"
    },
    {
```

```
        "Name": "requestparameters",
        "Type": "map<string,string>"
    },
    {
        "Name": "responseelements",
        "Type": "map<string,string>"
    },
    {
        "Name": "additionaleventdata",
        "Type": "map<string,string>"
    },
    {
        "Name": "requestid",
        "Type": "string"
    },
    {
        "Name": "eventid",
        "Type": "string"
    },
    {
        "Name": "readonly",
        "Type": "boolean"
    },
    {
        "Name": "resources",
        "Type": "array<struct<accountid:string,type:string,arn:string,arnprefix:string>>"
    },
    {
        "Name": "eventtype",
        "Type": "string"
    },
    {
        "Name": "apiversion",
        "Type": "string"
    },
    {
        "Name": "managementevent",
        "Type": "boolean"
    },
    {
        "Name": "recipientaccountid",
        "Type": "string"
    },
    {
        "Name": "sharedeventid",
        "Type": "string"
    },
    {
        "Name": "annotation",
        "Type": "string"
    },
    {
        "Name": "vpcendpointid",
        "Type": "string"
    },
    {
        "Name": "serviceeventdetails",
        "Type": "map<string,string>"
    },
    {
        "Name": "addendum",
        "Type": "map<string,string>"
    },
    {
        "Name": "edgedevicedetails",
        "Type": "map<string,string>"
```

```
    },
    {
        "Name": "insightdetails",
        "Type": "map<string,string>"
    },
    {
        "Name": "eventcategory",
        "Type": "string"
    },
    {
        "Name": "tlsdetails",
        "Type":
 "struct<tlsversion:string,ciphersuite:string,clientprovidedhostheader:string>"
    },
    {
        "Name": "sessioncredentialfromconsole",
        "Type": "string"
    },
    {
        "Name": "eventjson",
        "Type": "string"
    }
    {
        "Name": "eventjsonchecksum",
        "Type": "string"
    }
]
```

Supported schema for Amazon Config configuration item record fields

The following is the valid SQL schema for configuration item record fields. For configuration items, the value of `eventcategory` is `ConfigurationItem`, and the value of `eventtype` is `AwsConfigurationItem`.

```
[
    {
        "Name": "eventversion",
        "Type": "string"
    },
    {
        "Name": "eventcategory",
        "Type": "string"
    },
    {
        "Name": "eventtype",
        "Type": "string"
    },
        "Name": "eventid",
        "Type": "string"
    },
    {
        "Name": "eventtime",
        "Type": "timestamp"
    },
    {
        "Name": "awsregion",
        "Type": "string"
    },
    {
        "Name": "recipientaccountid",
        "Type": "string"
    },
```

```
    {
        "Name": "addendum",
        "Type": "map<string,string>"
    },
    {
        "Name": "eventdata",
        "Type": "struct<configurationitemversion:string,configurationitemcapturetime:

 string,configurationitemstatus:string,configurationitemstateid:string,accountid:string,

 resourcetype:string,resourceid:string,resourcename:string,arn:string,awsregion:string,

 availabilityzone:string,resourcecreationtime:string,configuration:map<string,string>,
                 supplementaryconfiguration:map<string,string>,relatedevents:string,
                 relationships:struct<name:string,resourcetype:string,resourceid:string,
                 resourcename:string>,tags:map<string,string>>"
    }
]
```

Supported schema for Amazon Audit Manager evidence record fields

The following is the valid SQL schema for Audit Manager evidence record fields. For Audit Manager evidence record fields, the value of `eventcategory` is `Evidence`, and the value of `eventtype` is `AwsAuditManagerEvidence`. For more information about aggregating evidence in CloudTrail Lake using Audit Manager, see Evidence finder in the *Amazon Audit Manager User Guide*.

```
[
    {
        "Name": "eventversion",
        "Type": "string"
    },
    {
        "Name": "eventcategory",
        "Type": "string"
    },
    {
        "Name": "eventtype",
        "Type": "string"
    },
        "Name": "eventid",
        "Type": "string"
    },
    {
        "Name": "eventtime",
        "Type": "timestamp"
    },
    {
        "Name": "awsregion",
        "Type": "string"
    },
    {
        "Name": "recipientaccountid",
        "Type": "string"
    },
    {
        "Name": "addendum",
        "Type": "map<string,string>"
    },
    {
        "Name": "eventdata",
        "Type":
 "struct<attributes:map<string,string>,awsaccountid:string,awsorganization:string,
```

```
 compliancecheck:string,datasource:string,eventname:string,eventsource:string,

 evidenceawsaccountid:string,evidencebytype:string,iamid:string,evidenceid:string,
                 time:timestamp,assessmentid:string,controlsetid:string,controlid:string,

 controlname:string,controldomainname:string,frameworkname:string,frameworkid:string,

 service:string,servicecategory:string,resourcearn:string,resourcetype:string,

 evidencefolderid:string,description:string,manualevidences3resourcepath:string,
                 evidencefoldername:string,resourcecompliancecheck:string>"
    }
]
```

Supported schema for non-Amazon event fields

The following is the valid SQL schema for non-Amazon events. For non-Amazon events, the value of `eventcategory` is `ActivityAuditLog`, and the value of `eventtype` is `ActivityLog`.

```
[
    {
        "Name": "eventversion",
        "Type": "string"
    },
    {
        "Name": "eventcategory",
        "Type": "string"
    },
    {
        "Name": "eventtype",
        "Type": "string"
    },
        "Name": "eventid",
        "Type": "string"
    },
    {
        "Name": "eventtime",
        "Type": "timestamp"
    },
    {
        "Name": "awsregion",
        "Type": "string"
    },
    {
        "Name": "recipientaccountid",
        "Type": "string"
    },
    {
        "Name": "addendum",
        "Type":
 "struct<reason:string,updatedfields:string,originalUID:string,originaleventid:string>"
    },
    {
        "Name": "metadata",
        "Type": "struct<ingestiontime:string,channelarn:string>"
    },
    {
        "Name": "eventdata",
        "Type": "struct<version:string,useridentity:struct<type:string,

 principalid:string,details:map<string,string>>,useragent:string,eventsource:string,

 eventname:string,eventtime:string,uid:string,requestparameters:map<string,string>>,
```

```
 responseelements":map<string,string>>,errorcode:string,errormssage:string,sourceipaddress:string,
                recipientaccountid:string,additionaleventdata":map<string,string>>"
    }
]
```

Supported CloudWatch metrics

CloudTrail Lake supports Amazon CloudWatch metrics. CloudWatch is a monitoring service for Amazon resources. You can use CloudWatch to collect and track metrics, set alarms, and automatically react to changes in your Amazon resources.

The `Amazon/CloudTrail` namespace includes the following metrics for CloudTrail Lake.

Metric	Description	Units
`HourlyDataIngested`	The amount of data ingested into the event data store during the last hour. This metric is updated every hour.	Bytes
`TotalDataRetained`	The amount of data retained in the event data store during its entire retention period. This metric is updated nightly.	Bytes

For more information about CloudWatch metrics, see the following topics.

- Using Amazon CloudWatch metrics
- Using Amazon CloudWatch alarms

Working with CloudTrail log files

You can perform more advanced tasks with your CloudTrail files.

- Create multiple trails per Region.
- Monitor CloudTrail log files by sending them to CloudWatch Logs.
- Share log files between accounts.
- Use the Amazon CloudTrail Processing Library to write log processing applications in Java.
- Validate your log files to verify that they have not changed after delivery by CloudTrail.

When an event occurs in your account, CloudTrail evaluates whether the event matches the settings for your trails. Only events that match your trail settings are delivered to your Amazon S3 bucket and Amazon CloudWatch Logs log group.

You can configure multiple trails differently so that the trails process and log only the events that you specify. For example, one trail can log read-only data and management events, so that all read-only events are delivered to one S3 bucket. Another trail can log only write-only data and management events, so that all write-only events are delivered to a separate S3 bucket.

You can also configure your trails to have one trail log and deliver all management events to one S3 bucket, and configure another trail to log and deliver all data events to another S3 bucket.

You can configure your trails to log the following:

- **Data events (p. 242)**: These events provide visibility into the resource operations performed on or within a resource. These are also known as data plane operations.
- **Management events (p. 238)**: Management events provide visibility into management operations that are performed on resources in your Amazon account. These are also known as control plane operations. Management events can also include non-API events that occur in your account. For example, when a user logs in to your account, CloudTrail logs the `ConsoleLogin` event. For more information, see Non-API events captured by CloudTrail (p. 430).

  Note
  Not all Amazon services support CloudTrail events. For more information about supported services, see CloudTrail supported services and integrations (p. 24). For specific details about what APIs are logged for a specific service, see that service's documentation in CloudTrail supported services and integrations (p. 24).
- **Insights events (p. 267)**: Insights events capture unusual activity that is detected in your account. If you have Insights events enabled, and CloudTrail detects unusual activity, Insights events are logged to the destination S3 bucket for your trail, but in a different folder. You can also see the type of Insights event and the incident time period when you view Insights events on the CloudTrail console. Unlike other types of events captured in a CloudTrail trail, Insights events are logged only when CloudTrail detects changes in your account's API usage that differ significantly from the account's typical usage patterns.

  Insights events are generated only for **write** management APIs.

Topics

- Create multiple trails (p. 237)
- Logging management events (p. 238)
- Logging data events (p. 242)

- Logging Insights events for trails (p. 267)
- Receiving CloudTrail log files from multiple Regions (p. 275)
- Managing data consistency in CloudTrail (p. 276)
- Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277)
- Receiving CloudTrail log files from multiple accounts (p. 290)
- Sharing CloudTrail log files between Amazon accounts (p. 297)
- Validating CloudTrail log file integrity (p. 307)
- Using the CloudTrail Processing Library (p. 327)

Create multiple trails

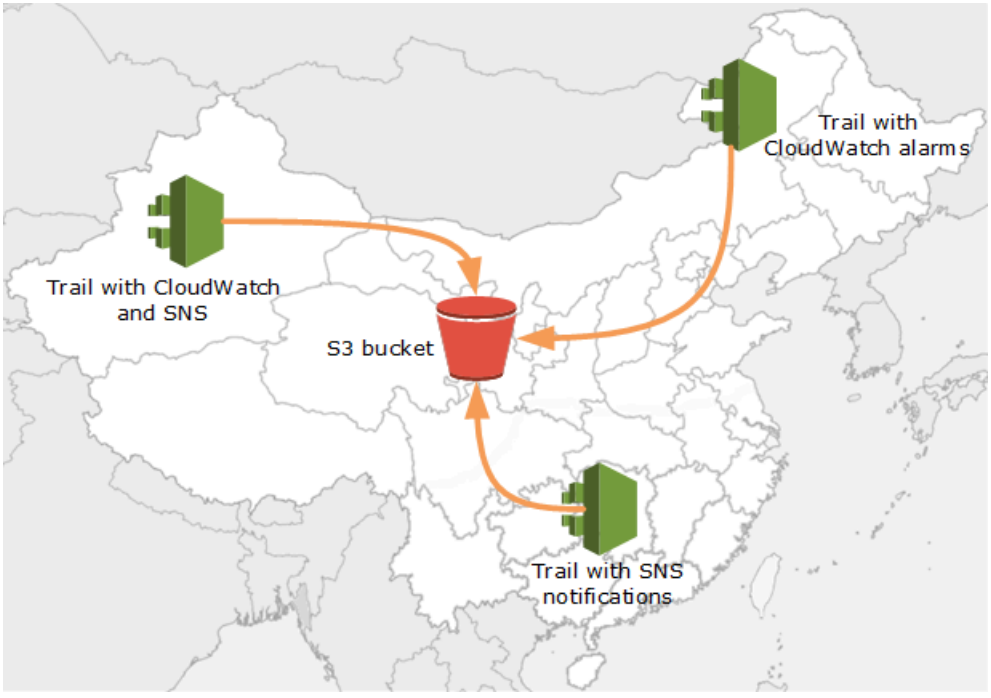
You can use CloudTrail log files to troubleshoot operational or security issues in your Amazon account. You can create trails for different users, who can create and manage their own trails. You can configure trails to deliver log files to separate S3 buckets or shared S3 buckets.

> **Note**
> Creating multiple trails will incur additional costs. For more information, see Amazon CloudTrail Pricing.

For example, you might have the following users:

- A security administrator creates a trail and configures SNS to receive notifications when new log files are delivered.
- A developer creates a trail and configures CloudWatch alarms to receive notifications for specific API activity.
- An IT auditor creates a trail and configures SNS and CloudWatch alarms.
- All log files are delivered to the same S3 bucket.

The following image illustrates this example.



Note
You can create up to five trails per Amazon Web Services Region A trail that logs activity from all Regions counts as one trail per Region.

You can use resource-level permissions to manage a user's ability to perform specific operations on CloudTrail.

For example, you might grant one user permission to view trail activity, but restrict the user from starting or stopping logging for a trail. You might grant another user full permission to create and delete trails. This gives you granular control over your trails and user access.

For more information about resource-level permissions, see Examples: Creating and applying policies for actions on specific trails (p. 352).

For more information about multiple trails, see the following resources:

- How does CloudTrail behave regionally and globally? (p. 13)
- CloudTrail FAQs

Logging management events

By default, trails log management events and don't include data or Insights events.

By default, event data stores log management events and don't include data events. Event data stores do not currently support Insights events.

Additional charges apply for data or Insights events. For more information, see Amazon CloudTrail Pricing.

Contents

- Management events (p. 238)
  - Logging management events with the Amazon Web Services Management Console (p. 239)
- Read and write events (p. 239)
- Logging events with the Amazon Command Line Interface (p. 240)
  - Examples: Logging management events for trails (p. 240)
- Logging events with the Amazon SDKs (p. 242)
- Sending events to Amazon CloudWatch Logs (p. 242)

Management events

Management events provide visibility into management operations that are performed on resources in your Amazon account. These are also known as control plane operations. Example management events include:

- Configuring security (for example, IAM `AttachRolePolicy` API operations)
- Registering devices (for example, Amazon EC2 `CreateDefaultVpc` API operations)
- Configuring rules for routing data (for example, Amazon EC2 `CreateSubnet` API operations)
- Setting up logging (for example, Amazon CloudTrail `CreateTrail` API operations)

Management events can also include non-API events that occur in your account. For example, when a user logs in to your account, CloudTrail logs the `ConsoleLogin` event. For more information, see Non-

API events captured by CloudTrail (p. 430). For a list of supported management events that CloudTrail logs for Amazon services, see CloudTrail supported services and integrations (p. 24).

By default, trails and event data stores are configured to log management events. For a list of supported management events that CloudTrail logs for Amazon services, see CloudTrail supported services and integrations (p. 24).

Note
The CloudTrail **Event history** feature supports only management events. Not all management events are supported in event history. You cannot exclude Amazon KMS events from **Event history** ; settings that you apply to a trail or event data store do not apply to **Event history** . For more information, see Viewing events with CloudTrail Event history (p. 50).

Logging management events with the Amazon Web Services Management Console

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. To update a trail, open the **Trails** page of the CloudTrail console and choose the trail name.

   To update an event data store, open the **Event data stores** page of the CloudTrail console and choose the event data store name.
3. For **Management events**, choose **Edit**.

   - Choose if you want your trail or event data store to log **Read** events, **Write** events, or both.
   - Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail or event data store. The default setting is to include all Amazon KMS events.

     The option to log or exclude Amazon KMS events is available only if you log management events on your trail or event data store. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

     Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

     To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

   Choose **Save changes** when you are finished.

Read and write events

When you configure your trail or event data store to log management events, you can specify whether you want read-only events, write-only events, or both.

- **Read**

  Read-only events include API operations that read your resources, but don't make changes. For example, read-only events include the Amazon EC2 `DescribeSecurityGroups` and `DescribeSubnets` API operations. These operations return only information about your Amazon EC2 resources and don't change your configurations.

- **Write**

  Write-only events include API operations that modify (or might modify) your resources. For example, the Amazon EC2 `RunInstances` and `TerminateInstances` API operations modify your instances.

Example: Logging read and write events for separate trails

The following example shows how you can configure trails to split log activity for an account into separate S3 buckets: one bucket receives read-only events and a second bucket receives write-only events.

1. You create a trail and choose an S3 bucket named `read-only-bucket` to receive log files. You then update the trail to specify that you want **Read** management events.
2. You create a second trail and choose an S3 bucket named `write-only-bucket` to receive log files. You then update the trail to specify that you want **Write** management events.
3. The Amazon EC2 `DescribeInstances` and `TerminateInstances` API operations occur in your account.
4. The `DescribeInstances` API operation is a read-only event and it matches the settings for the first trail. The trail logs and delivers the event to the `read-only-bucket`.
5. The `TerminateInstances` API operation is a write-only event and it matches the settings for the second trail. The trail logs and delivers the event to the `write-only-bucket`.

Logging events with the Amazon Command Line Interface

You can configure your trails to log management events using the Amazon CLI.

Examples: Logging management events for trails

To view whether your trail is logging management events, run the `get-event-selectors` command.

```
aws cloudtrail get-event-selectors --trail-name TrailName
```

The following example returns the default settings for a trail. By default, trails log all management events, log events from all event sources, and don't log data events.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:111122223333:trail/TrailName",
    "AdvancedEventSelectors": [
        {
            "Name": "Management events selector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Management"
                    ]
                }
            ]
        }
    ]
}
```

To configure your trail to log management events, run the `put-event-selectors` command. The following example shows how to configure your trail to include all management events for two S3

objects. You can specify from 1 to 5 event selectors for a trail. You can specify from 1 to 250 data resources for a trail.

Note
The maximum number of S3 data resources is 250, regardless of the number of event selectors.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-selectors
 '[{ "ReadWriteType": "All", "IncludeManagementEvents":true, "DataResources": [{ "Type":
 "AWS::S3::Object", "Values": ["arn:aws:s3:::mybucket/prefix", "arn:aws:s3:::mybucket2/
prefix2"] }] }]'
```

The following example returns the event selector configured for the trail.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:111122223333:trail/TrailName",
    "EventSelectors": [
        {
            "ReadWriteType": "All",
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Type": "AWS::S3::Object",
                    "Values": [
                        "arn:aws:s3:::mybucket/prefix",
                        "arn:aws:s3:::mybucket2/prefix2",
                    ]
                }
            ],
            "ExcludeManagementEventSources": []
        }
    ]
}
```

To exclude Amazon Key Management Service (Amazon KMS) events from a trail's logs, run the `put-event-selectors` command and add the attribute `ExcludeManagementEventSources` with a value of `kms.amazonaws.com`. The following example creates an event selector for a trail named *TrailName* to include read-only and write-only management events, but exclude Amazon KMS events. Because Amazon KMS can generate a high volume of events, the user in this example might want to limit events to manage the cost of a trail.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-
selectors '[{"ReadWriteType": "All","ExcludeManagementEventSources":
 ["kms.amazonaws.com"],"IncludeManagementEvents": true}]'
```

The example returns the event selector configured for the trail.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:111122223333:trail/TrailName",
    "EventSelectors": [
        {
            "ReadWriteType": "All",
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ExcludeManagementEventSources": [
                "kms.amazonaws.com"
            ]
        }
    ]
}
```

To start logging Amazon KMS events to a trail again, pass an empty string as the value of `ExcludeManagementEventSources`, as shown in the following command.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-selectors
 '[{"ReadWriteType": "All","ExcludeManagementEventSources": [],"IncludeManagementEvents":
 true}]'
```

To log relevant Amazon KMS events to a trail like `Disable`, `Delete` and `ScheduleKey`, but exclude high-volume Amazon KMS events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, log write-only management events, and keep the default setting to log Amazon KMS events, as shown in the following example.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-
selectors '[{"ReadWriteType": "WriteOnly","ExcludeManagementEventSources":
 [],"IncludeManagementEvents": true}]'
```

Logging events with the Amazon SDKs

Use the GetEventSelectors operation to see whether your trail is logging management events for a trail. You can configure your trails to log management events with the PutEventSelectors operation. For more information, see the Amazon CloudTrail API Reference.

Sending events to Amazon CloudWatch Logs

CloudTrail supports sending data and management events to CloudWatch Logs. When you configure your trail to send events to your CloudWatch Logs log group, CloudTrail sends only the events that you specify in your trail. For example, if you configure your trail to log management events only, your trail delivers management events only to your CloudWatch Logs log group. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

Logging data events

By default, trails and event data stores do not log data events. Additional charges apply for data events. For more information, see Amazon CloudTrail Pricing.

> **Note**
> The events that are logged by your trails are available in Amazon EventBridge. For example, if you choose to log data events for S3 objects but not management events, your trail processes and logs only data events for the specified S3 objects. The data events for these S3 objects are available in Amazon EventBridge. For more information, see Events from Amazon services in the *Amazon EventBridge User Guide*.

Contents

- Data events (p. 243)
  - Examples: Logging data events for Amazon S3 objects (p. 251)
  - Logging data events for S3 objects in other Amazon accounts (p. 252)
- Read-only and write-only events (p. 253)
- Logging data events with the Amazon Command Line Interface (p. 254)
  - Examples: Logging data events for trails (p. 254)
    - Log events by using basic event selectors (p. 254)
    - Log events by using advanced event selectors (p. 255)

- Log all Amazon S3 events for a bucket by using advanced event selectors (p. 257)
- Log Amazon S3 on Amazon Outposts events by using advanced event selectors (p. 257)
- Log all data events by using advanced event selectors (p. 258)
- Logging events with the Amazon SDKs (p. 267)
- Sending events to Amazon CloudWatch Logs (p. 267)

Data events

Data events provide visibility into the resource operations performed on or within a resource. These are also known as data plane operations. Data events are often high-volume activities.

The following table shows the data event types available for trails. The **Data event type (console)** column shows the appropriate selection in the console. The **resources.Type** column shows the `resources.Type` value that you would specify to include data events of that type in your trail.

For trails, you can use basic or advanced event selectors to log data events for Amazon S3 buckets and bucket objects, Lambda functions, and DynamoDB tables (shown in the first three rows of the table). You can use only advanced event selectors to log the data event types shown in the remaining rows.

Amazon service	Description	Data event type (console)	resources.Type
Amazon DynamoDB	Amazon DynamoDB object-level API activity on tables (for example, `PutItem`, `DeleteItem`, and `UpdateItem` API operations). For more information about DynamoDB events, see DynamoDB data plane events in CloudTrail.	DynamoDB	`AWS::DynamoDB::Table`
Amazon Lambda	Amazon Lambda function execution activity (the `Invoke` API).	Lambda	`AWS::Lambda::Function`
Amazon S3	Amazon S3 object-level API activity (for example, `GetObject`, `DeleteObject`, and `PutObject` API operations) on buckets and objects in buckets.	S3	`AWS::S3::Object`
Amazon CloudTrail	CloudTrail `PutAuditEvents` activity on a CloudTrail Lake channel (p. 180) that is used to log events from outside Amazon.	CloudTrail	`AWS::CloudTrail::Channel`

Amazon service	Description	Data event type (console)	resources.Type
Amazon CodeWhisperer	Amazon CodeWhisperer API activity on a profile.	CodeWhisperer	`AWS::CodeWhisperer::Profile`
Amazon Cognito	Amazon Cognito API activity on Amazon Cognito identity pools.	Cognito Identity Pools	`AWS::Cognito::IdentityPool`
Amazon DynamoDB	Amazon DynamoDB API activity on streams.	DynamoDB Streams	`AWS::DynamoDB::Stream`
Amazon Elastic Block Store	Amazon Elastic Block Store (EBS) direct APIs, such as `PutSnapshotBlock`, `GetSnapshotBlock`, and `ListChangedBlocks` on Amazon EBS snapshots.	Amazon EBS direct APIs	`AWS::EC2::Snapshot`
Amazon EMR	Amazon EMR API activity on a write-ahead log workspace.	EMR write-ahead log workspace	`AWS::EMRWAL::Workspace`
Amazon FinSpace	Amazon FinSpace API activity on environments.	FinSpace	`AWS::FinSpace::Environment`
Amazon Glue	Amazon Glue API activity on tables that were created by Lake Formation. **Note** Amazon Glue data events for tables are currently supported only in the following regions: • US East (N. Virginia) • US East (Ohio) • US West (Oregon) • Europe (Ireland) • Asia Pacific (Tokyo) Region	Lake Formation	`AWS::Glue::Table`

Amazon service	Description	Data event type (console)	resources.Type
Amazon GuardDuty	Amazon GuardDuty API activity for a detector.	GuardDuty detector	`AWS::GuardDuty::Detector`
Amazon HealthImaging	Amazon HealthImaging API activity on data stores.	Medical Imaging data store	`AWS::MedicalImaging::Datastore`
Amazon Kendra Intelligent Ranking	Amazon Kendra Intelligent Ranking API activity on rescore execution plans.	Kendra Ranking	`AWS::KendraRanking::ExecutionPlan`
Amazon Managed Blockchain	Amazon Managed Blockchain API activity on a network.	Managed Blockchain network	`AWS::ManagedBlockchain::Network`
Amazon Managed Blockchain	Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, such as `eth_getBalance` or `eth_getBlockByNumber`.	Managed Blockchain	`AWS::ManagedBlockchain::Node`
Amazon SageMaker	Amazon SageMaker API activity on feature stores.	SageMaker feature store	`AWS::SageMaker::FeatureGroup`
Amazon SageMaker	Amazon SageMaker API activity on experiment trial components.	SageMaker metrics experiment trial component	`AWS::SageMaker::ExperimentTrialComponent`
Amazon S3	Amazon S3 API activity on access points.	S3 Access Point	`AWS::S3::AccessPoint`
Amazon S3	Amazon S3 Object Lambda access points API activity, such as calls to `CompleteMultipartUpload` and `GetObject`.	S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`
Amazon S3 on Outposts	Amazon S3 on Outposts object-level API activity.	S3 Outposts	`AWS::S3Outposts::Object`
Amazon Systems Manager	Amazon Systems Manager API activity on control channels.	Systems Manager	`AWS::SSMMessages::ControlChannel`
Amazon Verified Permissions	Amazon Verified Permissions API activity on a policy store.	Amazon Verified Permissions	`AWS::VerifiedPermissions::PolicyStore`

Data events are not logged by default when you create a trail or event data store. To record CloudTrail data events, you must explicitly add the supported resources or resource types for which you want to collect activity. For more information, see Creating a trail (p. 78) and Create an event data store (p. 159).

On a single-Region trail or event data store, you can log data events only for resources that you can access in that Region. Though S3 buckets are global, Amazon Lambda functions and DynamoDB tables are regional.

Additional charges apply for logging data events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

Steps for logging data events depend on whether you have advanced event selectors enabled. You can use only advanced event selectors for event data stores. For trails, you can use either basic or advanced event selectors, but not both. Use the procedure in this section that matches the kind of event selectors you have enabled.

Logging data events with basic event selectors in the Amazon Web Services Management Console

Use the following procedure to log data events using basic event selectors.

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. Open the **Trails** page of the CloudTrail console and choose the trail name.

   Note
   While you can edit an existing trail to log data events, as a best practice, consider creating a separate trail specifically for logging data events.
3. For **Data events**, choose **Edit**.
4. For Amazon S3 buckets:

   a. For **Data event source**, choose **S3**.

   b. You can choose to log **All current and future S3 buckets**, or you can specify individual buckets or functions. By default, data events are logged for all current and future S3 buckets.

      Note
      Keeping the default **All current and future S3 buckets** option enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
      If you are creating a trail for a single Region (done by using the Amazon CLI), selecting the **Select all S3 buckets in your account** option enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.

   c. If you leave the default, **All current and future S3 buckets**, choose to log **Read** events, **Write** events, or both.

   d. To select individual buckets, empty the **Read** and **Write** check boxes for **All current and future S3 buckets**. In **Individual bucket selection**, browse for a bucket on which to log data events. To find specific buckets, type a bucket prefix for the bucket you want. You can select multiple buckets in this window. Choose **Add bucket** to log data events for more buckets. Choose to log **Read** events, such as `GetObject`, **Write** events, such as `PutObject`, or both.

      This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.

To remove a bucket from logging, choose **X**.

5. To add another data type on which to log data events, choose **Add data event type**.
6. For Lambda functions:

   a. For **Data event source**, choose **Lambda**.

   b. In **Lambda function**, choose **All regions** to log all Lambda functions, or **Input function as ARN** to log data events on a specific function.

      To log data events for all Lambda functions in your Amazon account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.

      Note
      If you are creating a trail for all Regions, this selection enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
      Logging data events for all functions also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

   c. If you choose **Input function as ARN**, enter the ARN of a Lambda function.

      Note
      If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still select the option to log all functions, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

7. To add another data type on which to log data events, choose **Add data event type**.
8. For DynamoDB tables:

   a. For **Data event source**, choose **DynamoDB**.

   b. In **DynamoDB table selection**, choose **Browse** to select a table, or paste in the ARN of a DynamoDB table to which you have access. A DynamoDB table ARN uses the following format:

      ```
      arn:partition:dynamodb:region:account_ID:table/table_name
      ```

      To add another table, choose **Add row**, and browse for a table or paste in the ARN of a table to which you have access.

9. Choose **Save changes**.

Logging data events with advanced event selectors in the Amazon Web Services Management Console

In the Amazon Web Services Management Console, if you have advanced event selectors enabled, you can choose from predefined templates that log all data events on a selected resource (Amazon S3 buckets or access points, Lambda functions, S3 objects on Amazon Outposts, Ethereum for Managed Blockchain nodes, or S3 Object Lambda access points). After you choose a log selector template, you can customize the template to include only the data events you most want to see. For more information and

tips about using advanced event selectors, see Log events by using advanced event selectors (p. 255) in this topic.

1. On the **Dashboard**, **Trails**, or **Event data stores** pages of the CloudTrail console, choose the trail or event data store you want to update.

   Note
   You can only enable data events on event data stores that contain CloudTrail events. You cannot enable data events on CloudTrail event data stores for Amazon Config configuration items or non-Amazon events.

2. On the details page, in **Data events**, choose **Edit**.
3. If you are not already logging data events, choose the **Data events** check box.
4. For **Data event type**, choose the resource type on which you want to log data events.
5. Choose a log selector template. CloudTrail includes predefined templates that log all data events for the resource type. To build a custom log selector template, choose **Custom**.

   Note
   Choosing a predefined template for S3 buckets enables data event logging for all buckets currently in your Amazon account and any buckets you create after you finish creating the trail or event data store. It also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
   If the trail or event data store applies only to one Region, choosing a predefined template that logs all S3 buckets enables data event logging for all buckets in the same Region as your trail or event data store and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.
   If you are creating a trail or event data store for all Regions, choosing a predefined template for Lambda functions enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail or event data store. If you are creating a trail or event data store for a single Region (for trails, this only can be done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail or event data store. It does not enable data event logging for Lambda functions created in other Regions. Logging data events for all functions also enables logging of data event activity performed by any user or role in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

6. (Optional) In **Selector name**, enter a name to identify your selector. The selector name is a descriptive name for an advanced event selector, such as "Log data events for only two S3 buckets". The selector name is listed as `Name` in the advanced event selector and is viewable if you expand the **JSON view**.
7. In **Advanced event selectors**, build an expression for the specific resources on which you want to log data events. You can skip this step if you are using a predefined log template.

   a. Choose from the following fields. For fields that accept an array (more than one value), CloudTrail adds an OR between values.

      - **`readOnly`** - `readOnly` can be set to **Equals** a value of `true` or `false`. Read-only data events are events that do not change the state of a resource, such as `Get*` or `Describe*` events. Write events add, change, or delete resources, attributes, or artifacts, such as `Put*`, `Delete*`, or `Write*` events. To log both `read` and `write` events, don't add a `readOnly` selector.
      - **`eventName`** - `eventName` can use any operator. You can use it to include or exclude any data event logged to CloudTrail, such as `PutBucket`, `GetItem`, or `GetSnapshotBlock`. You can have multiple values for this field, separated by commas.

- **`resources.ARN`** - You can use any operator with `resources.ARN`, but if you use **Equals** or **NotEquals**, the value must exactly match the ARN of a valid resource of the type you've specified in the template as the value of `resources.type`.

  The following table shows the valid ARN format for each `resources.Type`.

resources.Type	resources.ARN
`AWS::DynamoDB::Table`	`arn:`*`partition`*`:dynamodb:`*`region`*`:`*`account_ID`*`:table/`*`table_name`*
`AWS::Lambda::Function`	`arn:`*`partition`*`:lambda:`*`region`*`:`*`account_ID`*`:function:`*`function_nar`
`AWS::S3::Object`[1]	`arn:`*`partition`*`:s3:::`*`bucket_name`*`/` `arn:`*`partition`*`:s3:::`*`bucket_name`*`/` *`object_or_file_name`*`/`
`AWS::CloudTrail::Channel`	`arn:`*`partition`*`:cloudtrail:`*`region`*`:`*`account_ID`*`:channel/` *`channel_UUID`*
`AWS::CodeWhisperer::Profile`	`arn:`*`partition`*`:codewhisperer:`*`region`*`:`*`account_ID`*`:profile/` *`profile_ID`*

`AWS::Cognito::IdentityPool`	`arn:`*`partition`*`:cognito-`
	`identity:`*`region`*`:`*`account_ID`*`:identitypool/`
	`identity_pool_ID`

`AWS::DynamoDB::Stream`	`arn:`*`partition`*`:dynamodb:`*`region`*`:`*`account_ID`*`:table/` *`table_name`*`/stream/`*`date_time`*
`AWS::EC2::Snapshot`	`arn:`*`partition`*`:ec2:`*`region`*`::snapshot/`*`snapshot_ID`*
`AWS::EMRWAL::Workspace`	`arn:`*`partition`*`:emrwal:`*`region`*`::workspace/` *`workspace_name`*
`AWS::FinSpace::Environment`	`arn:`*`partition`*`:finspace:`*`region`*`:`*`account_ID`*`:environment/` *`environment_ID`*
`AWS::Glue::Table`	`arn:`*`partition`*`:glue:`*`region`*`:`*`account_ID`*`:table/` *`database_name`*`/`*`table_name`*
`AWS::GuardDuty::Detector`	`arn:`*`partition`*`:guardduty:`*`region`*`:`*`account_ID`*`:detector/` *`detector_ID`*

`AWS::KendraRanking::ExecutionPlan`	`arn:`*`partition`*`:kendra-`
	`ranking:`*`region`*`:`*`account_ID`*`:rescore-execution-`
	`plan/`*`rescore_execution_plan_ID`*

resources.Type	resources.ARN
AWS::ManagedBlockchain::Network	arn:*partition*:managedblockchain:::networks/*network_name*
AWS::ManagedBlockchain::Node	arn:*partition*:managedblockchain:*region*:*account_ID*:nodes/*node_ID*
AWS::MedicalImaging::Datastore	arn:*partition*:medical-imaging:*region*:*account_ID*:datastore/*data_store_ID*
AWS::SageMaker::FeatureGroup	arn:*partition*:sagemaker:*region*:*account_ID*:feature-group/*feature_group_name*
AWS::SageMaker::ExperimentTrialComponent	arn:*partition*:sagemaker:*region*:*account_ID*:experiment-trial-component/*experiment_trial_component_name*
AWS::S3::AccessPoint[2]	arn:*partition*:s3:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3ObjectLambda::AccessPoint	arn:*partition*:s3-object-lambda:*region*:*account_ID*:accesspoint/*access_point_name*
AWS::S3Outposts::Object	arn:*partition*:s3-outposts:*region*:*account_ID*:*object_path*
AWS::SSMMessages::ControlChannel	arn:*partition*:ssmmessages:*region*:*account_ID*:control-channel/*control_channel_ID*
AWS::VerifiedPermissions::PolicyStore	arn:*partition*:verifiedpermissions:*region*:*account_ID*:policy-store/*policy_store_ID*

[1] To log all data events for all objects in a specific S3 bucket, use the `StartsWith` operator, and include only the bucket ARN as the matching value. The trailing slash is intentional; do not exclude it.

[2] To log events on all objects in an S3 access point, we recommend that you use only the access point ARN, don't include the object path, and use the `StartsWith` or `NotStartsWith` operators.

For more information about the ARN formats of data event resources, see Actions, resources, and condition keys in the *Amazon Identity and Access Management User Guide*.

b. For each field, choose **+ Conditions** to add as many conditions as you need, up to a maximum of 500 specified values for all conditions. For example, to exclude data events for two S3 buckets from data events that are logged on your trail or event data store, you can set the field to

resources.ARN, set the operator for **NotStartsWith**, and then either paste in an S3 bucket ARN, or browse for the S3 buckets for which you do not want to log events.

To add the second S3 bucket, choose **+ Conditions**, and then repeat the preceding instruction, pasting in the ARN for or browsing for a different bucket.

Note
You can have a maximum of 500 values for all selectors on a trail or event data store. This includes arrays of multiple values for a selector such as `eventName`. If you have single values for all selectors, you can have a maximum of 500 conditions added to a selector.
If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail or event data store. You can still log all functions with a predefined selector template, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail or event data store in the console, and then use the Amazon CLI to configure data event logging for specific Lambda functions. For more information, see Logging data events with the Amazon Command Line Interface (p. 254).

c. Choose **+ Field** to add additional fields as required. To avoid errors, do not set conflicting or duplicate values for fields. For example, do not specify an ARN in one selector to be equal to a value, then specify that the ARN not equal the same value in another selector.

d. Save changes to your custom selector template by choosing **Next**. Do not choose another log selector template, or leave this page, or your custom selectors will be lost.

8. To add another data type on which to log data events, choose **Add data event type**. Repeat steps 4 through this step to configure advanced event selectors for the data event type.
9. After you choose **Next**, in **Step 2: Choose log events**, review the log selector template options you've chosen. Choose **Edit** to go back and make changes.
10. After you've reviewed and verified your choices, choose **Save changes** if this is an existing trail or event data store, or **Create trail** or **Create event data store** if you are creating a new trail or event data store.

Examples: Logging data events for Amazon S3 objects

Logging data events for all S3 objects in an S3 bucket

The following example demonstrates how logging works when you configure logging of all data events for an S3 bucket named *`bucket-1`*. In this example, the CloudTrail user specified an empty prefix, and the option to log both **Read** and **Write** data events.

1. A user uploads an object to `bucket-1`.
2. The `PutObject` API operation is an Amazon S3 object-level API. It is recorded as a data event in CloudTrail. Because the CloudTrail user specified an S3 bucket with an empty prefix, events that occur on any object in that bucket are logged. The trail processes and logs the event.
3. Another user uploads an object to `bucket-2`.
4. The `PutObject` API operation occurred on an object in an S3 bucket that wasn't specified for the trail. The trail doesn't log the event.

Logging data events for specific S3 objects

The following example demonstrates how logging works when you configure a trail to log events for specific S3 objects. In this example, the CloudTrail user specified an S3 bucket named *`bucket-3`*, with the prefix *`my-images`*, and the option to log only **Write** data events.

1. A user deletes an object that begins with the `my-images` prefix in the bucket, such as `arn:aws:s3:::bucket-3/my-images/example.jpg`.
2. The `DeleteObject` API operation is an Amazon S3 object-level API. It is recorded as a **Write** data event in CloudTrail. The event occurred on an object that matches the S3 bucket and prefix specified in the trail. The trail processes and logs the event.
3. Another user deletes an object with a different prefix in the S3 bucket, such as `arn:aws:s3:::bucket-3/my-videos/example.avi`.
4. The event occurred on an object that doesn't match the prefix specified in your trail. The trail doesn't log the event.
5. A user calls the `GetObject` API operation for the object, `arn:aws:s3:::bucket-3/my-images/example.jpg`.
6. The event occurred on a bucket and prefix that are specified in the trail, but `GetObject` is a read-type Amazon S3 object-level API. It is recorded as a **Read** data event in CloudTrail, and the trail is not configured to log **Read** events. The trail doesn't log the event.

   Note
   If you are logging data events for specific Amazon S3 buckets, we recommend you do not use an Amazon S3 bucket for which you are logging data events to receive log files that you have specified in the data events section. Using the same Amazon S3 bucket causes your trail or event data store to log a data event each time log files are delivered to your Amazon S3 bucket. Log files are aggregated events delivered at intervals, so this is not a 1:1 ratio of event to log file; the event is logged in the next log file. For example, when CloudTrail delivers logs, the `PutObject` event occurs on the S3 bucket. If the S3 bucket is also specified in the data events section, the trail processes and logs the `PutObject` event as a data event. That action is another `PutObject` event, and the trail processes and logs the event again. For more information, see How CloudTrail works (p. 1).
   To avoid logging data events for the Amazon S3 bucket where you receive log files if you configure a trail to log all Amazon S3 data events in your Amazon account, consider configuring delivery of log files to an Amazon S3 bucket that belongs to another Amazon account. For more information, see Receiving CloudTrail log files from multiple accounts (p. 290).

Logging data events for S3 objects in other Amazon accounts

When you configure your trail to log data events, you can also specify S3 objects that belong to other Amazon accounts. When an event occurs on a specified object, CloudTrail evaluates whether the event matches any trails in each account. If the event matches the settings for a trail, the trail processes and logs the event for that account. Generally, both API callers and resource owners can receive events.

If you own an S3 object and you specify it in your trail, your trail logs events that occur on the object in your account. Because you own the object, your trail also logs events when other accounts call the object.

If you specify an S3 object in your trail, and another account owns the object, your trail only logs events that occur on that object in your account. Your trail doesn't log events that occur in other accounts.

Example: Logging data events for an Amazon S3 object for two Amazon accounts

The following example shows how two Amazon accounts configure CloudTrail to log events for the same S3 object.

1. In your account, you want your trail to log data events for all objects in your S3 bucket named `owner-bucket`. You configure the trail by specifying the S3 bucket with an empty object prefix.
2. Bob has a separate account that has been granted access to the S3 bucket. Bob also wants to log data events for all objects in the same S3 bucket. For his trail, he configures his trail and specifies the same S3 bucket with an empty object prefix.
3. Bob uploads an object to the S3 bucket with the `PutObject` API operation.

4. This event occurred in his account and it matches the settings for his trail. Bob's trail processes and logs the event.
5. Because you own the S3 bucket and the event matches the settings for your trail, your trail also processes and logs the same event. Because there are now two copies of the event (one logged in Bob's trail, and one logged in yours), CloudTrail charges for two copies of the data event.
6. You upload an object to the S3 bucket.
7. This event occurs in your account and it matches the settings for your trail. Your trail processes and logs the event.
8. Because the event didn't occur in Bob's account, and he doesn't own the S3 bucket, Bob's trail doesn't log the event. CloudTrail charges for only one copy of this data event.

Example: Logging data events for all buckets, including an S3 bucket used by two Amazon accounts

The following example shows the logging behavior when **Select all S3 buckets in your account** is enabled for trails that collect data events in an Amazon account.

1. In your account, you want your trail to log data events for all S3 buckets. You configure the trail by choosing **Read** events, **Write** events, or both for **All current and future S3 buckets** in **Data events**.
2. Bob has a separate account that has been granted access to an S3 bucket in your account. He wants to log data events for the bucket to which he has access. He configures his trail to get data events for all S3 buckets.
3. Bob uploads an object to the S3 bucket with the `PutObject` API operation.
4. This event occurred in his account and it matches the settings for his trail. Bob's trail processes and logs the event.
5. Because you own the S3 bucket and the event matches the settings for your trail, your trail also processes and logs the event. Because there are now two copies of the event (one logged in Bob's trail, and one logged in yours), CloudTrail charges each account for a copy of the data event.
6. You upload an object to the S3 bucket.
7. This event occurs in your account and it matches the settings for your trail. Your trail processes and logs the event.
8. Because the event didn't occur in Bob's account, and he doesn't own the S3 bucket, Bob's trail doesn't log the event. CloudTrail charges for only one copy of this data event in your account.
9. A third user, Mary, has access to the S3 bucket, and runs a `GetObject` operation on the bucket. She has a trail configured to log data events on all S3 buckets in her account. Because she is the API caller, CloudTrail logs a data event in her trail. Though Bob has access to the bucket, he is not the resource owner, so no event is logged in his trail this time. As the resource owner, you receive an event in your trail about the `GetObject` operation that Mary called. CloudTrail charges your account and Mary's account for each copy of the data event: one in Mary's trail, and one in yours.

Read-only and write-only events

When you configure your trail or event data store to log data and management events, you can specify whether you want read-only events, write-only events, or both.

- **Read**

  Read events include API operations that read your resources, but don't make changes. For example, read-only events include the Amazon EC2 `DescribeSecurityGroups` and `DescribeSubnets` API operations. These operations return only information about your Amazon EC2 resources and don't change your configurations.
- **Write**

Write events include API operations that modify (or might modify) your resources. For example, the Amazon EC2 `RunInstances` and `TerminateInstances` API operations modify your instances.

Example: Logging read and write events for separate trails

The following example shows how you can configure trails to split log activity for an account into separate S3 buckets: one bucket receives read-only events and a second bucket receives write-only events.

1. You create a trail and choose an S3 bucket named `read-only-bucket` to receive log files. You then update the trail to specify that you want **Read** management events and data events.
2. You create a second trail and choose an S3 bucket named `write-only-bucket` to receive log files. You then update the trail to specify that you want **Write** management events and data events.
3. The Amazon EC2 `DescribeInstances` and `TerminateInstances` API operations occur in your account.
4. The `DescribeInstances` API operation is a read-only event and it matches the settings for the first trail. The trail logs and delivers the event to the `read-only-bucket`.
5. The `TerminateInstances` API operation is a write-only event and it matches the settings for the second trail. The trail logs and delivers the event to the `write-only-bucket`.

Logging data events with the Amazon Command Line Interface

You can configure your trails or event data stores to log data events using the Amazon CLI.

Examples: Logging data events for trails

You can configure your trails to log management and data events using the Amazon CLI. To see whether your trail is logging management and data events, run the `get-event-selectors` command.

Note
Be aware that if your account is logging more than one copy of management events, you incur charges. There is always a charge for logging data events. For more information, see Amazon CloudTrail Pricing.

```
aws cloudtrail get-event-selectors --trail-name TrailName
```

The command returns the default settings for a trail.

Topics

- Log events by using basic event selectors (p. 254)
- Log events by using advanced event selectors (p. 255)
- Log all Amazon S3 events for a bucket by using advanced event selectors (p. 257)
- Log Amazon S3 on Amazon Outposts events by using advanced event selectors (p. 257)
- Log all data events by using advanced event selectors (p. 258)

Log events by using basic event selectors

The following is an example result of the **get-event-selectors** command showing basic event selectors. By default, when you create a trail by using the Amazon CLI, a trail logs all management events. By default, trails do not log data events.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName",
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ReadWriteType": "All"
        }
    ]
}
```

To configure your trail to log management and data events, run the `put-event-selectors` command.

The following example shows how to use basic event selectors to configure your trail to include all management and data events for the S3 objects in two S3 bucket prefixes. You can specify from 1 to 5 event selectors for a trail. You can specify from 1 to 250 data resources for a trail.

Note
The maximum number of S3 data resources is 250, if you choose to limit data events by using basic event selectors.

```
aws cloudtrail put-event-selectors --trail-name TrailName --event-selectors
 '[{ "ReadWriteType": "All", "IncludeManagementEvents":true, "DataResources": [{ "Type":
 "AWS::S3::Object", "Values": ["arn:aws:s3:::mybucket/prefix", "arn:aws:s3:::mybucket2/
prefix2"] }] }]'
```

The command returns the event selectors that are configured for the trail.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName",
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [
                {
                    "Values": [
                        "arn:aws:s3:::mybucket/prefix",
                        "arn:aws:s3:::mybucket2/prefix2",
                    ],
                    "Type": "AWS::S3::Object"
                }
            ],
            "ReadWriteType": "All"
        }
    ]
}
```

Log events by using advanced event selectors

If you have opted to use advanced event selectors, the **get-event-selectors** command returns results similar to the following. By default, no advanced event selectors are configured for a trail.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName",
    "AdvancedEventSelectors": []

}
```

The following example shows how to use advanced event selectors to log all management events (both `readOnly` and `writeOnly`), and include `PutObject` and `DeleteObject` events for the S3 objects in

the same two S3 bucket prefixes. As shown here, you can use advanced event selectors to select not only the S3 prefix names by ARN, but the names of the specific events that you want to log. You can add up to 500 conditions to advanced event selectors per trail, including all selector values. You can specify from 1 to 250 data resources for a trail.

```
aws cloudtrail put-event-selectors --trail-name TrailName \
--advanced-event-selectors
'[
  {
    "Name": "Log readOnly and writeOnly management events",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Management"] }
    ]
  },
  {
    "Name": "Log PutObject and DeleteObject events for two S3 prefixes",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3::Object"] },
      { "Field": "eventName", "Equals": ["PutObject","DeleteObject"] },
      { "Field": "resources.ARN", "StartsWith": ["arn:aws:s3:::mybucket/
prefix","arn:aws:s3:::mybucket2/prefix2"] }
    ]
  }
]'
```

The result shows the advanced event selectors that are configured for the trail.

```
{
  "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName",
  "AdvancedEventSelectors": [
    {
      "Name": "Log readOnly and writeOnly management events",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
                "Management"
           ]
        }
      ]
    },
    {
      "Name": "Log PutObject and DeleteObject events for two S3 prefixes",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
                "Data"
           ]
        },
        {
          "Field": "resources.type",
          "Equals": [
                "AWS::S3::Object"
           ]
        },
        {
          "Field": "resources.ARN",
          "StartsWith": [
                "arn:aws:s3:::mybucket/prefix","arn:aws:s3:::mybucket2/prefix2"
           ]
        }
      ]
```

```
    }
  ]
}
```

Log all Amazon S3 events for a bucket by using advanced event selectors

The following example shows how to configure your trail to include all data events for all Amazon S3 objects in a specific S3 bucket. The value for S3 events for the `resources.type` field is `AWS::S3::Object`. Because the ARN values for S3 objects and S3 buckets are slightly different, you must add the `StartsWith` operator for `resources.ARN` to capture all events.

```
aws cloudtrail put-event-selectors --trail-name TrailName --region region \
--advanced-event-selectors \
'[
    {
            "Name": "S3EventSelector",
            "FieldSelectors": [
                { "Field": "eventCategory", "Equals": ["Data"] },
                { "Field": "resources.type", "Equals": ["AWS::S3::Object"] },
                { "Field": "resources.ARN", "StartsWith":
 ["arn:partition:s3:::bucket_name/"] }
            ]
        }
]'
```

The command returns the following example output.

```
{
    "TrailARN": "arn:aws:cloudtrail:region:account_ID:trail/TrailName",
    "AdvancedEventSelectors": [
        {
            "Name": "S3EventSelector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                },
                {
                    "Field": "resources.ARN",
                    "StartsWith": [
                        "arn:partition:s3:::bucket_name/"
                    ]
                }
            ]
        }
    ]
}
```

Log Amazon S3 on Amazon Outposts events by using advanced event selectors

The following example shows how to configure your trail to include all data events for all Amazon S3 on Outposts objects in your outpost.

```
aws cloudtrail put-event-selectors --trail-name TrailName --region region \
--advanced-event-selectors \
'[
    {
            "Name": "OutpostsEventSelector",
            "FieldSelectors": [
                { "Field": "eventCategory", "Equals": ["Data"] },
                { "Field": "resources.type", "Equals": ["AWS::S3Outposts::Object"] }
            ]
    }
]'
```

The command returns the following example output.

```
{
    "TrailARN": "arn:aws:cloudtrail:region:account_ID:trail/TrailName",
    "AdvancedEventSelectors": [
        {
            "Name": "OutpostsEventSelector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3Outposts::Object"
                    ]
                }
            ]
        }
    ]
}
```

Log all data events by using advanced event selectors

The following example shows how to configure your trail to include data events for all S3 buckets, Amazon Lambda functions, `PutAuditEvents` calls on CloudTrail Lake channels, DynamoDB tables, Amazon S3 on Outposts, Amazon Managed Blockchain JSON-RPC calls on Ethereum nodes, S3 Object Lambda access points, Amazon EBS direct APIs on EBS snapshots, S3 access points, DynamoDB streams, Amazon Glue tables created by Lake Formation, Amazon FinSpace environments, Amazon SageMaker metrics experiment trial components, Amazon SageMaker feature stores, Amazon Kendra rescore execution plans, Amazon Cognito identity pools, Amazon GuardDuty detectors, Amazon EMR write-ahead log workspaces, Amazon CodeWhisperer profiles, Amazon Verified Permissions policy stores, Amazon Systems Manager control channels, Amazon Managed Blockchain networks, and Amazon HealthImaging data stores.

> **Note**
> If the trail applies only to one Region, only events in that Region are logged, even though the event selector parameters specify all Amazon S3 buckets and Lambda functions. In a single-Region trail, event selectors apply only to the Region where the trail is created.

```
aws cloudtrail put-event-selectors --trail-name TrailName \
--advanced-event-selectors '
[
  {
    "Name": "Log all events for all Amazon S3 buckets",
```

```
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3::Object"] }
    ]
  },
  {
    "Name": "Log all events for Lambda functions",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Lambda::Function"] }
    ]
  },
  {
    "Name": "Log all events for DynamoDB tables",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::DynamoDB::Table"] }
    ]
  },
  {
    "Name": "Log all CloudTrail PutAuditEvents activity on a CloudTrail Lake channel",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::CloudTrail::Channel"] }
    ]
  },
  {
    "Name": "Log all events for Amazon S3 on Outposts",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3Outposts::Object"] }
    ]
  },
  {
    "Name": "Log all JSON-RPC calls for Ethereum nodes in Amazon Managed Blockchain",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::ManagedBlockchain::Node"] }
    ]
  },
  {
    "Name": "Log all events for Amazon S3 Object Lambda access points",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3ObjectLambda::AccessPoint"] }
    ]
  },
  {
    "Name": "Log all Amazon EBS direct API calls on snapshots",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::EC2::Snapshot"] }
    ]
  },
  {
    "Name": "Log all events for Amazon S3 access points",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::S3::AccessPoint"] }
    ]
  },
  {
    "Name": "Log all events for DynamoDB streams",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::DynamoDB::Stream"] }
```

```
    ]
  },
  {
    "Name": "Log all events for Amazon Glue tables created by Lake Formation",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Glue::Table"] }
    ]
  },
  {
    "Name": "Log all events for FinSpace environments",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::FinSpace::Environment"] }
    ]
  },
  {
    "Name": "Log all events for SageMaker metrics experiment trial components",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::SageMaker::ExperimentTrialComponent"] }
    ]
  },
  {
    "Name": "Log all events for SageMaker feature stores",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::SageMaker::FeatureGroup"] }
    ]
  },
  {
    "Name": "Log all events for Amazon Kendra Intelligent Ranking rescore execution plans",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::KendraRanking::ExecutionPlan"] }
    ]
  },
  {
    "Name": "Log all events for Amazon Cognito identity pools",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::Cognito::IdentityPool"] }
    ]
  },
  {
    "Name": "Log all events for an Amazon GuardDuty detector",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::GuardDuty::Detector"] }
    ]
  },
  {
    "Name": "Log all events for Amazon EMR write-ahead log workspaces",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::EMRWAL::Workspace"] }
    ]
  },
  {
    "Name": "Log all events for Amazon CodeWhisperer profiles",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::CodeWhisperer::Profile"] }
     ]
  },
  {
```

```
    "Name": "Log all events for Amazon Verified Permissions policy stores",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::VerifiedPermissions::PolicyStore"] }
     ]
  },
  {
    "Name": "Log all events for Amazon Systems Manager control channels",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::SSMMessages::ControlChannel"] }
     ]
  },
  {
    "Name": "Log all events for Amazon Managed Blockchain networks",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::ManagedBlockchain::Network"] }
     ]
  },
  {
    "Name": "Log all events for Amazon HealthImaging data stores",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::MedicalImaging::Datastore"] }
     ]
  }
]'
```

The command returns the following example output.

```
{
    "TrailARN": "arn:aws:cloudtrail:us-east-1:111122223333:trail/TrailName",
    "AdvancedEventSelectors": [
        {
            "Name": "Log all events for all Amazon S3 buckets",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3::Object"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Lambda functions",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Lambda::Function"
                    ]
```

```
                }
            ]
        },
        {
            "Name": "Log all events for DynamoDB tables",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::DynamoDB::Table"
                    ]
                }
            ]
        },
        {
            "Name": "Log all CloudTrail PutAuditEvents activity on a CloudTrail Lake
channel",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::CloudTrail::Channel"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon S3 on Outposts",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::S3Outposts::Object"
                    ]
                }
            ]
        },
        {
            "Name": "Log all JSON-RPC calls for Ethereum nodes in Amazon Managed
Blockchain",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
```

```
                "Field": "resources.type",
                "Equals": [
                    "AWS::ManagedBlockchain::Node"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for Amazon S3 Object Lambda access points",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::S3ObjectLambda::AccessPoint"
                ]
            }
        ]
    },
    {
        "Name": "Log all Amazon EBS direct API calls on snapshots",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::EC2::Snapshot"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for Amazon S3 access points",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::S3::AccessPoint"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for DynamoDB streams",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
```

```
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::DynamoDB::Stream"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for Amazon Glue tables created by Lake Formation",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::Glue::Table"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for FinSpace environments",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::FinSpace::Environment"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for SageMaker metrics experiment trial components",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
                    "Data"
                ]
            },
            {
                "Field": "resources.type",
                "Equals": [
                    "AWS::SageMaker::ExperimentTrialComponent"
                ]
            }
        ]
    },
    {
        "Name": "Log all events for SageMaker feature stores",
        "FieldSelectors": [
            {
                "Field": "eventCategory",
                "Equals": [
```

```
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SageMaker::FeatureGroup"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Kendra Intelligent Ranking rescore execution
plans",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::KendraRanking::ExecutionPlan"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Cognito identity pools",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::Cognito::IdentityPool"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for an Amazon GuardDuty detector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::GuardDuty::Detector"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon EMR write-ahead log workspaces",
            "FieldSelectors": [
```

```
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::EMRWAL::Workspace"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon CodeWhisperer profiles",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::CodeWhisperer::Profile"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Verified Permissions policy stores",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::VerifiedPermissions::PolicyStore"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon Systems Manager control channels",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::SSMMessages::ControlChannel"
                    ]
                }
            ]
        },
        {
```

```
            "Name": "Log all events for Amazon Managed Blockchain networks",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::ManagedBlockchain::Network"
                    ]
                }
            ]
        },
        {
            "Name": "Log all events for Amazon HealthImaging data stores",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Data"
                    ]
                },
                {
                    "Field": "resources.type",
                    "Equals": [
                        "AWS::MedicalImaging::Datastore"
                    ]
                }
            ]
        }
    ]
}
```

Logging events with the Amazon SDKs

Run the GetEventSelectors operation to see whether your trail is logging data events. You can configure your trails to log data events by running the PutEventSelectors operation. For more information, see the Amazon CloudTrail API Reference.

Sending events to Amazon CloudWatch Logs

CloudTrail supports sending data events to CloudWatch Logs. When you configure your trail to send events to your CloudWatch Logs log group, CloudTrail sends only the events that you specify in your trail. For example, if you configure your trail to log data events only, your trail delivers data events only to your CloudWatch Logs log group. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

Logging Insights events for trails

Amazon CloudTrail Insights helps Amazon users identify and respond to unusual activity associated with API calls and API error rates by continuously analyzing CloudTrail management events. CloudTrail Insights analyzes your normal patterns of API call volume and API error rates, also called the *baseline*, and generates Insights events when the call volume or error rates are outside normal patterns. Insights events on API call volume are generated for `write` management APIs, and Insights events on API error rate are generated for both `read` and `write` management APIs.

Note
To log Insights events on API call volume, the trail must log `write` management events. To log Insights events on API error rate, the trail must log `read` or `write` management events.

If you have CloudTrail Insights enabled and CloudTrail detects unusual activity, Insights events are delivered to the destination S3 bucket for your trail. You can also see the type of insight and the incident time period when you view Insights events on the CloudTrail console. Unlike other types of events captured in a CloudTrail trail, Insights events are logged only when CloudTrail detects changes in your account's API usage that differ significantly from the account's typical usage patterns.

CloudTrail Insights continuously monitors CloudTrail management events, and uses mathematical models to determine the normal levels of API event and error rate activity for an account. CloudTrail Insights identifies behavior that is outside normal patterns, generates Insights events, and delivers those events to a `/CloudTrail-Insight` folder in the chosen destination S3 bucket for your trail. You can also access and view Insights events in the Amazon Web Services Management Console for CloudTrail. For more information about how to access and view Insights events in the console and by using the Amazon CLI, see Viewing CloudTrail Insights events (p. 62) in this guide.

By default, trails log management events and don't include data events or Insights events. Additional charges apply for data and Insights events. For more information, see Amazon CloudTrail Pricing.

When an event occurs in your account, CloudTrail evaluates whether the event matches the settings for your trails. Only events that match your trail settings are delivered to your Amazon S3 bucket and Amazon CloudWatch Logs log group.

Contents

- Understanding CloudTrail Insights (p. 268)
  - Filter column (p. 269)
  - Insights graph tab (p. 269)
  - Attributions tab (p. 271)
    - Baseline average and Insights average (p. 272)
  - CloudTrail events tab (p. 273)
  - Insights event record tab (p. 273)
- Logging Insights events with the Amazon Web Services Management Console (p. 273)
- Logging Insights events with the Amazon Command Line Interface (p. 273)
- Logging events with the Amazon SDKs (p. 274)
- Sending events to Amazon CloudWatch Logs (p. 274)

Understanding CloudTrail Insights

CloudTrail Insights can help you detect unusual API volume or error rate activity in your Amazon account by raising Insights events. CloudTrail Insights analyzes your normal patterns of API call volume and API error rates, also called the *baseline*, and generates Insights events when the call volume or error rates are outside normal patterns. Insights events on API call volume are generated for `write` management APIs, and Insights events on API error rate are generated for both `read` and `write` management APIs.

After you enable CloudTrail Insights for the first time on a trail, it can take up to 36 hours for CloudTrail to deliver the first Insights event, if unusual activity is detected. CloudTrail Insights analyzes management events that occur in a single Region, not globally. A CloudTrail Insights event is generated in the same Region as its supporting management events are generated.

The following image shows an example of Insights events. You open details pages for an Insights event by choosing an Insights event name from the **Dashboard** or **Insights** pages.

If you disable CloudTrail Insights on a trail, or stop logging on a trail (which disables CloudTrail Insights), you may have Insights events stored in your destination S3 bucket, or shown on the **Insights** page of the console, that date from the earlier time that you had Insights enabled.

Filter column

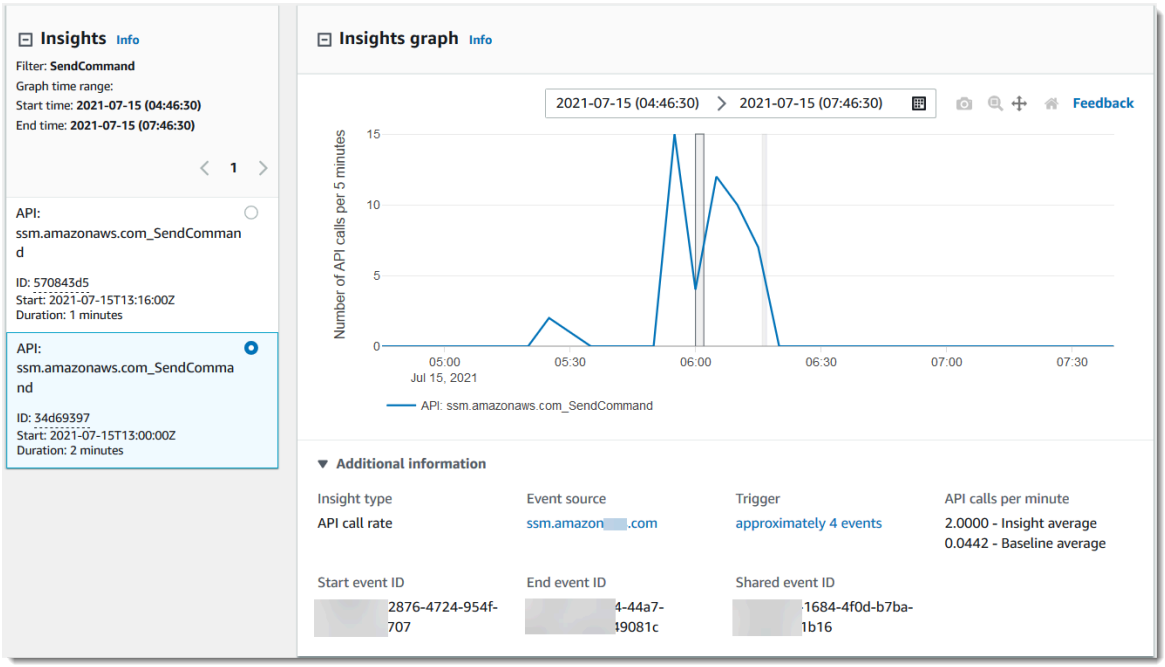
The left column lists Insights events that are related to the subject API, and that have the same Insights event type. The column lets you choose the Insights event about which you want more information. When you choose an event in this column, the event is highlighted in the graph on the **Insights graph** tab. By default, CloudTrail applies a filter that limits events shown on the **CloudTrail events** tab to those about the specific API that was called during the period of unusual activity that triggered the Insights event. To show all CloudTrail events called during the period of unusual activity, including events unrelated to the Insights event, turn off the filter.

Insights graph tab

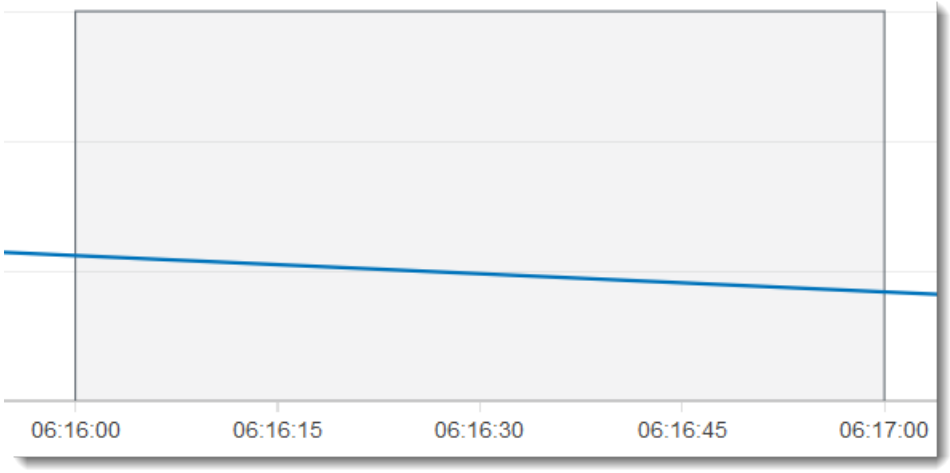
On the **Insights graph** tab, the details page for an Insights event shows a graph of an API's call volume or error rate that occurred over a period of time before and after one or more Insights events are logged. In the graph, Insights events are highlighted with vertical bars, with the width of the bar showing the start and end time of the Insights event.

In this example, a vertical highlighting band shows unusual numbers of Amazon Systems Manager SendCommand API calls in an account. In the highlighted area, because the number of SendCommand calls rose above the account's baseline average of 0.0442 calls per minute, CloudTrail logged an Insights event when it detected the unusual activity. The Insights event recorded that as many as 15 SendCommand calls were made in a five-minute period between 5:50 and 5:55 a.m. This is about two more calls to that API per minute than is expected for the account. In this example, the graph's time span is three hours: 4:30 a.m. PDT on July 15, 2021 to 7:30 a.m. PDT on July 15, 2021. This event has a start time of 6:00 a.m. PDT on July 15, 2021, and an end time two minutes later. An ending Insights event, not highlighted, shows that the unusual activity ended at about 6:16 a.m.

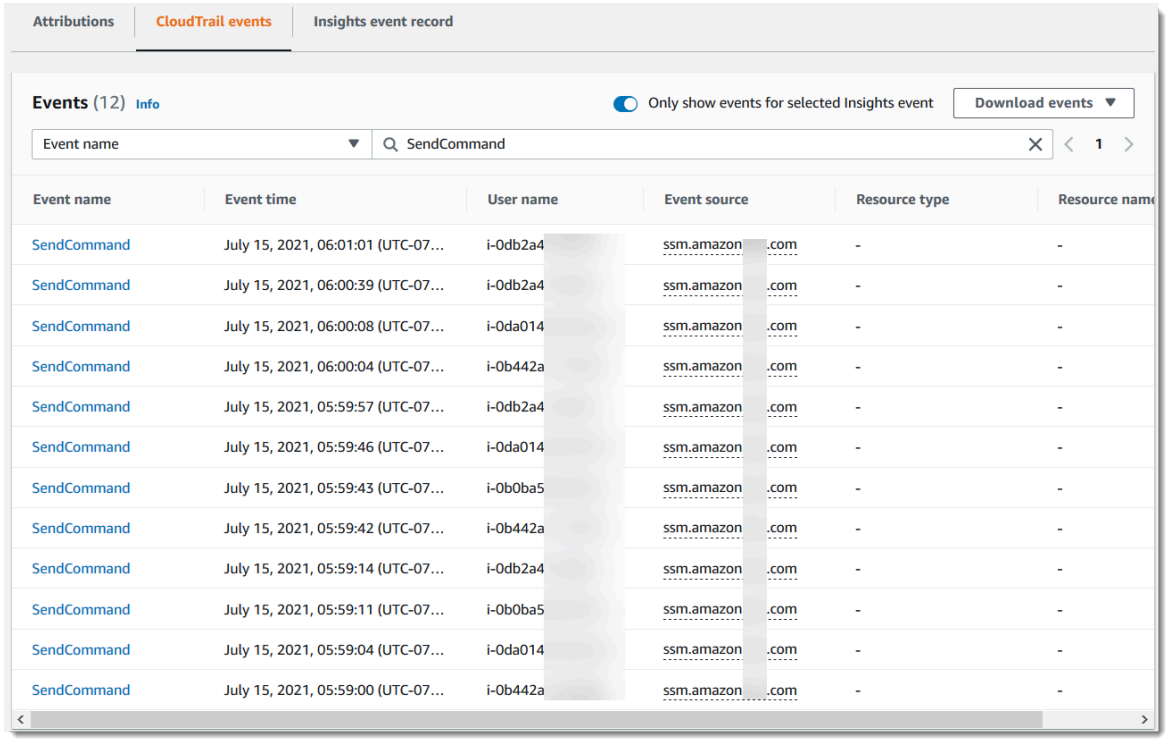
The baseline is calculated over the seven days preceding the start of an Insights event. Though the value of the baseline duration—the period that CloudTrail analyzes for normal activity on APIs—is approximately seven days, CloudTrail rounds the baseline duration to a whole integer day, so the exact baseline duration can vary.



You can use the **Zoom** command on the toolbar to zoom in on the ending Insights event, showing the start and end time. In this example, choosing **Zoom**, then dragging the **Zoom** cursor a very short distance over one edge of the highlighted Insights event expands the Insights event and shows more timeline detail.

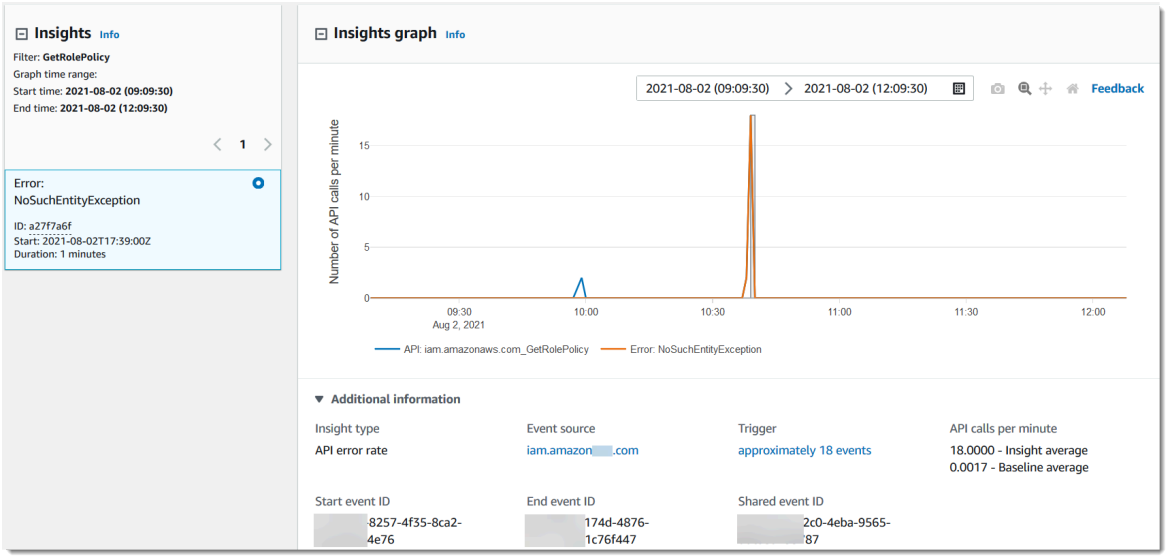


To view CloudTrail events that were analyzed to determine unusual activity, open the **CloudTrail events** tab. In this example, CloudTrail analyzed 12 events, four of which triggered the Insights event.



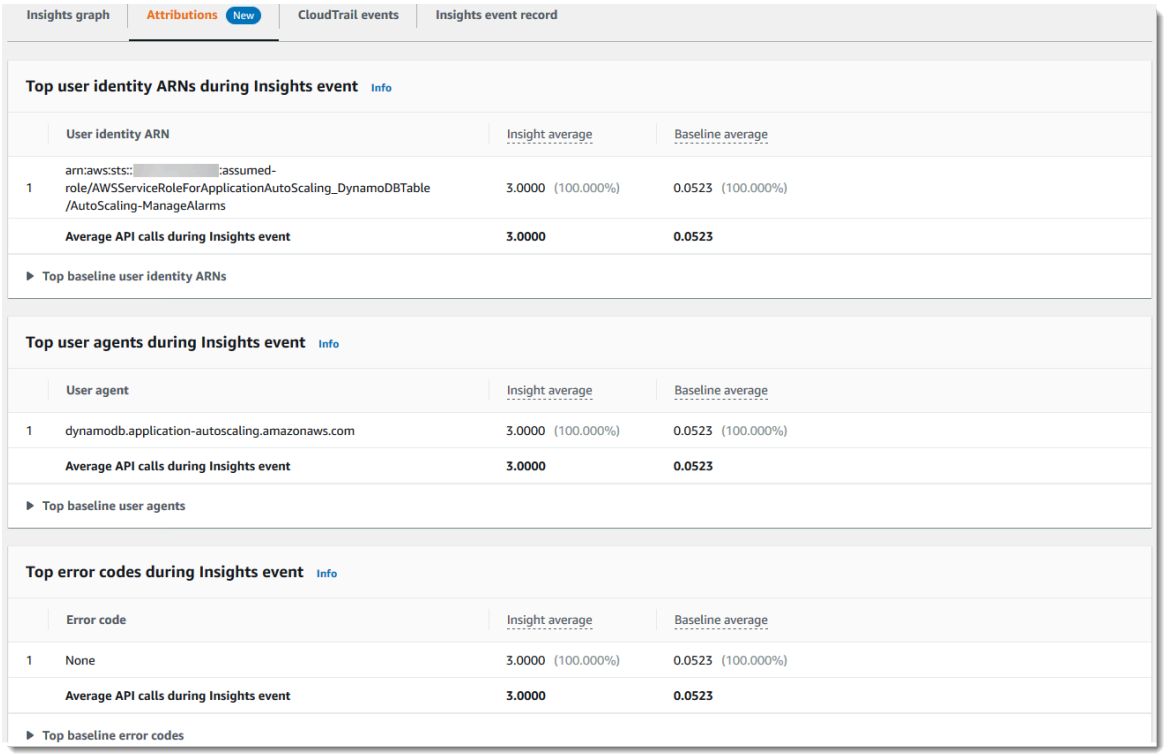
The following screenshot shows an Insights graph tab for an API error rate Insights event. The highlighted area shows that an Insights event was logged because occurrences of the `NoSuchEntityException` error on the `GetRolePolicy` IAM API call rose above the baseline average of 0.0017 `NoSuchEntityException` errors per minute on this API call, averaging 18 errors per minute during the insight period. The number of CloudTrail events that triggered the Insights event matches the Insights average of 18 `NoSuchEntityException` errors in one minute, in this example. Unlike an API call rate graph, the API error rate shows two lines, in contrasting colors: a line measuring calls to the IAM

API, `GetRolePolicy`, that resulted in an unusual number of errors, and a line measuring the error on which unusual activity was logged, `NoSuchEntityException`.

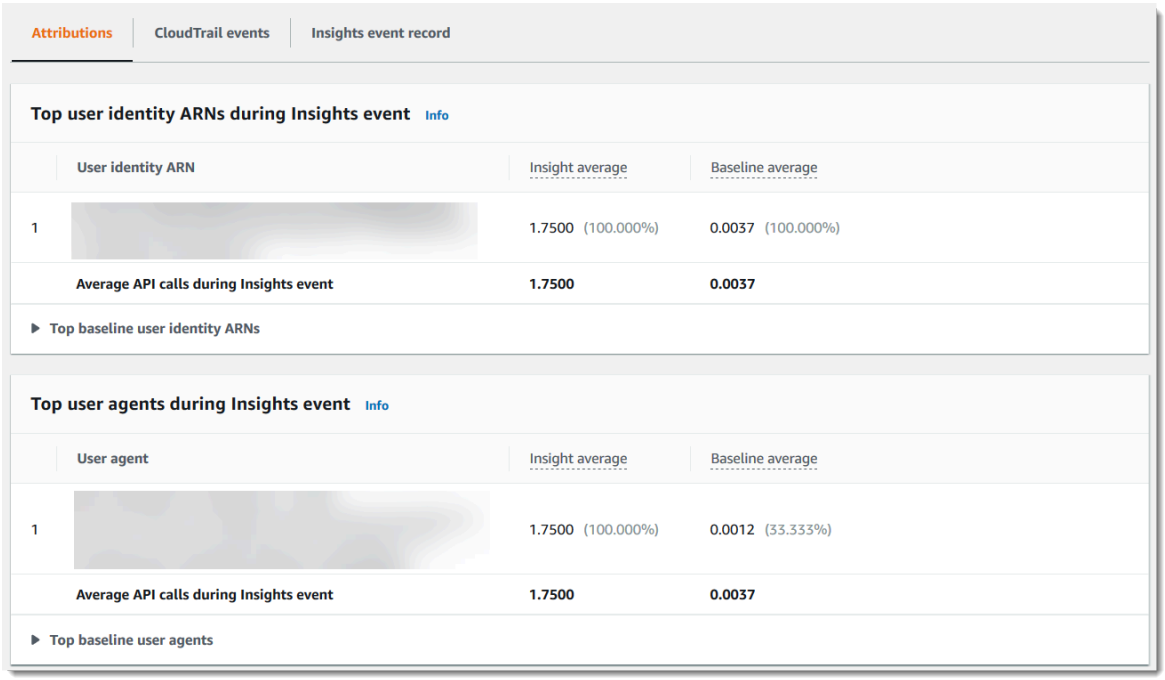


Attributions tab

The **Attributions** tab shows the following information about an Insights event. Information on the **Attributions** tab can help you identify the causes and sources of Insights activity. Expand the top baseline areas to compare user identity, user agent, and error code activity during normal periods with those attributed during the Insights activity. In **Top baseline user identity ARNs**, **Top baseline user agents**, and **Top baseline error codes**, only the *baseline average*—the historic average of events for the API that are logged by the user identity, user agent, or that result in the error code, in approximately the seven days before the Insights event start time—is shown.



The **Attributions** tab shows only top user identity ARNs and top user agents for an error rate Insights event, as shown in the following image. Top error codes are not necessary for error rate Insights events.



- **Top user identity ARNs** - This table shows up to the top five Amazon users or IAM roles (user identities) that contributed to API calls during the unusual activity and baseline periods, in descending order by the average number of API calls contributed. The percentage of the averages as a total of activity that contributed to the unusual activity is shown in parentheses. If more than five user identity ARNs contributed to the unusual activity, their activity is summed up in an **Other** row.
- **Top user agents** - This table shows up to the top five Amazon tools by which the user identity contributed to API calls during the unusual activity and baseline periods, in descending order by the average number of API calls contributed. These tools include the Amazon Web Services Management Console, Amazon CLI, or the Amazon SDKs. For example, a user agent named `ec2.amazonaws.com` indicates that the Amazon EC2 console was among the tools used to call the API. The percentage of the averages as a total of activity that contributed to the unusual activity is shown in parentheses. If more than five user agents contributed to the unusual activity, their activity is summed up in an **Other** row.
- **Top error codes** - Only shown for **API call rate** Insights events. This table shows up to the top five error codes that occurred on API calls during the unusual activity and baseline periods, in descending order from largest number of API calls to smallest. The percentage of the averages as a total of activity that contributed to the unusual activity is shown in parentheses. If more than five error codes occurred during the unusual or baseline activity, their activity is summed up in an **Other** row.

  A value of None as one of the top five error code values means that a significant percentage of the calls that contributed to the Insights event did not result in errors. If the error code value is None, and there are no other error codes in the table, the values in the **Insight average** and **Baseline average** columns are the same as those for the Insights event overall. You can also see those values displayed in the **Insight average** and **Baseline average** legend on the **Insights graph** tab, under **API calls per minute**.

Baseline average and Insights average

Baseline average and **Insights average** are shown for top user identities, top user agents, and top error codes.

- **Baseline average** - The typical rate of occurrences per minute on the API on which the Insights event was logged, as measured within approximately the preceding seven days, in a specific Region in your account.
- **Insights average** - The rate of calls to or errors on this API that triggered the Insights event. The CloudTrail Insights average for the start event is the rate of calls or errors per minute on the API that triggered the Insights event. Typically, this is the first minute of unusual activity. The Insights average for the end event is the rate of API calls or errors per minute over the duration of the unusual activity, between the start Insights event and the end Insights event.

CloudTrail events tab

On the **CloudTrail events** tab, view related events that CloudTrail analyzed to determine that unusual activity occurred. By default, a filter is already applied for the Insights event name, which is also the name of the related API. To show all CloudTrail events logged during the period of unusual activity, turn off **Only show events for selected Insights event**. The **CloudTrail events** tab shows CloudTrail management events related to the subject API that occurred between the start and end time of the Insights event. These events help you perform deeper analysis to determine the probable cause of an Insights event, and reasons for unusual API and error rate activity.

Insights event record tab

Like any CloudTrail event, a CloudTrail Insights event is a record in JSON format. The **Insights event record** tab shows the JSON structure and content of the Insights start and end events, sometimes called the event *payload*. For more information about the fields and content of the Insights event record, see Record fields for Insights events (p. 409) and CloudTrail Insights `insightDetails` element (p. 424) in this guide.

Logging Insights events with the Amazon Web Services Management Console

Enable CloudTrail Insights events on an existing trail. By default, Insights events are not enabled.

1. In the left navigation pane of the CloudTrail console, open the **Trails** page, and choose a trail name.
2. In **Insights events** choose **Edit**.

   Note
   Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.
3. In **Event type**, choose **Insights events**.
4. In **Insights events**, under **Choose Insights types**, choose **API call rate**, **API error rate**, or both. Your trail must be logging **Write** management events to log Insights events for **API call rate**. Your trail must be logging **Read** or **Write** management events to log Insights events for **API error rate**.
5. Choose **Save changes** to save your changes.

It can take up to 36 hours for CloudTrail to deliver the first Insights events, if unusual activity is detected.

Logging Insights events with the Amazon Command Line Interface

You can configure your trails to log Insights events using the Amazon CLI.

Note
To log Insights events on API call volume, the trail must log `write` management events. To log Insights events on API error rate, the trail must log `read` or `write` management events.

To view whether your trail is logging Insights events, run the `get-insight-selectors` command.

```
aws cloudtrail get-insight-selectors --trail-name TrailName
```

The following result shows the default settings for a trail. By default, trails don't log Insights events. The `InsightType` attribute value is empty, and no Insight event selectors are specified, because Insights event collection is not enabled.

If you do not add Insights selectors, the **get-insight-selectors** command returns the following error message: "An error occurred (InsightNotEnabledException) when calling the GetInsightSelectors operation: Trail *name* does not have Insights enabled. Edit the trail settings to enable Insights, and then try the operation again."

```
{
  "InsightSelectors": [ ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/TrailName"
}
```

To configure your trail to log Insights events, run the `put-insight-selectors` command. The following example shows how to configure your trail to include Insights events. Insights selector values can be `ApiCallRateInsight`, `ApiErrorRateInsight`, or both.

```
aws cloudtrail put-insight-selectors --trail-name TrailName --insight-selectors
 '[{"InsightType": "ApiCallRateInsight"},{"InsightType": "ApiErrorRateInsight"}]'
```

The following result shows the Insights event selector that is configured for the trail.

```
{
  "InsightSelectors":
      [
         {
            "InsightType": "ApiErrorRateInsight"
         },
         {
            "InsightType": "ApiCallRateInsight"
         }
      ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/TrailName"
}
```

Logging events with the Amazon SDKs

Run the GetInsightSelectors operation to see whether your trail is logging Insights events for a trail. You can configure your trails to log Insights events with the PutInsightSelectors operation. For more information, see the Amazon CloudTrail API Reference.

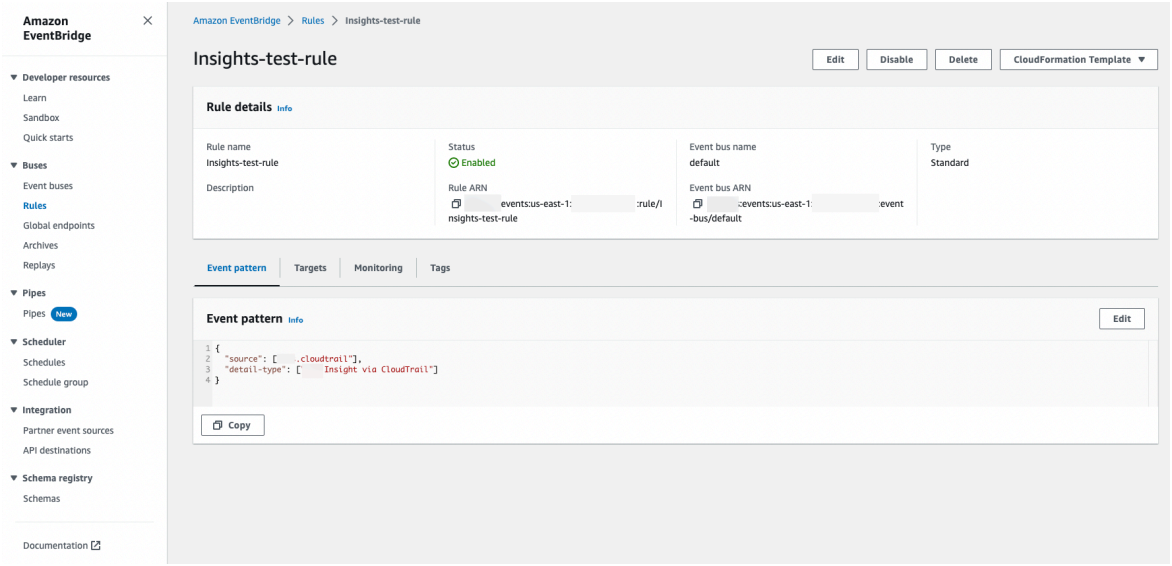
Sending events to Amazon CloudWatch Logs

CloudTrail supports sending Insights events to CloudWatch Logs. When you configure your trail to send Insights events to your CloudWatch Logs log group, CloudTrail Insights sends only the events that you

specify in your trail. For example, if you configure your trail to log management and Insights events, your trail delivers management and Insights events to your CloudWatch Logs log group. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

You can use the Amazon EventBridge console or API to create a rule to deliver Insights events. When you create a rule using the EventBridge console, choose the `AWS Insight via CloudTrail` event type.

The following image shows an example rule, **Insights-test-rule**, created in EventBridge. When CloudTrail logs Insights events, the rule targets an Amazon SNS topic to send notifications to recipients who are specified in the SNS topic. For more information about creating rules in EventBridge, see Create a rule in Amazon EventBridge in the *Amazon EventBridge User Guide*.



When CloudTrail logs Insights events, recipients of the SNS topic should receive SNS notifications.

Receiving CloudTrail log files from multiple Regions

You can configure CloudTrail to deliver log files from multiple Regions to a single S3 bucket for a single account. For example, you have a trail in the US West (Oregon) Region that is configured to deliver log files to a S3 bucket, and a CloudWatch Logs log group. When you change an existing single-Region trail to log all Regions, CloudTrail logs events from all Regions that are in a single Amazon partition in your account. CloudTrail delivers log files to the same S3 bucket and CloudWatch Logs log group. As long as CloudTrail has permissions to write to an S3 bucket, the bucket for a multi-Region trail does not have to be in the trail's home Region.

To log events across all Regions in all Amazon partitions in your account, create a multi-Region trail in each partition.

In the console, by default, you create a trail that logs events in all Amazon Web Services Regions. This is a recommended best practice. To log events in a single Region (not recommended), use the Amazon CLI (p. 100). To configure an existing single-Region trail to log in all Regions, you must use the Amazon CLI.

To change an existing trail so that it applies to all Regions, add the `--is-multi-region-trail` option to the **update-trail** (p. 100) command.

```
aws cloudtrail update-trail --name my-trail --is-multi-region-trail
```

To confirm that the trail now applies to all Regions, the `IsMultiRegionTrail` element in the output shows `true`.

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "my-trail",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "S3BucketName": "my-bucket"
}
```

Note
When a new Region launches in the aws-cn partition, CloudTrail automatically creates a trail in the new Region with the same settings as your original trail.

For more information, see the following resources:

- How does CloudTrail behave regionally and globally? (p. 13)
- CloudTrail FAQs

Managing data consistency in CloudTrail

CloudTrail uses a distributed computing model called eventual consistency. Any change that you make to your CloudTrail configuration (or other Amazon services), including tags used in attribute-based access control (ABAC), takes time to become visible from all possible endpoints. Some of the delay results from the time it takes to send the data from server to server, from replication zone to replication zone, and from Region to Region around the world. CloudTrail also uses caching to improve performance, but in some cases this can add time. The change might not be visible until the previously cached data times out.

You must design your applications to account for these potential delays. Ensure that they work as expected, even when a change made in one location is not instantly visible at another. Such changes include creating or updating trails or event data stores, updating event selectors, and starting or stopping logging. When you create or update a trail or event data store, CloudTrail delivers logs to the S3 bucket or event data store based on the last known configuration until the changes propagate to all locations.

For more information about how this affects other Amazon Web Services, see the following resources:

- **Amazon DynamoDB**: What is the consistency model of DynamoDB? in the *DynamoDB FAQ*, and Read consistency in the *Amazon DynamoDB Developer Guide*.
- **Amazon EC2**: Eventual consistency in the *Amazon Elastic Compute Cloud API Reference*.
- **Amazon EMR**: Ensuring Consistency When Using Amazon S3 and Amazon Elastic MapReduce for ETL Workflows in the *Amazon Big Data Blog*.
- **Amazon Identity and Access Management (IAM)**: Changes that I make are not always immediately visible in the *IAM User Guide*.
- **Amazon Redshift**: Managing data consistency in the *Amazon Redshift Database Developer Guide*.
- **Amazon S3**: Amazon S3 data consistency model in the *Amazon Simple Storage Service User Guide*.

Monitoring CloudTrail Log Files with Amazon CloudWatch Logs

You can configure CloudTrail with CloudWatch Logs to monitor your trail logs and be notified when specific activity occurs.

1. Configure your trail to send log events to CloudWatch Logs.
2. Define CloudWatch Logs metric filters to evaluate log events for matches in terms, phrases, or values. For example, you can monitor for `ConsoleLogin` events.
3. Assign CloudWatch metrics to the metric filters.
4. Create CloudWatch alarms that are triggered according to thresholds and time periods that you specify. You can configure alarms to send notifications when alarms are triggered, so that you can take action.
5. You can also configure CloudWatch to automatically perform an action in response to an alarm.

Standard pricing for Amazon CloudWatch and Amazon CloudWatch Logs applies. For more information, see Amazon CloudWatch Pricing.

For more information about the Regions in which you can configure your trails to send logs to CloudWatch Logs, see Amazon CloudWatch Logs Regions and Quotas in the *Amazon General Reference*.

The China (Beijing) Region requires a separate account. For more information, see China (Beijing).

Topics

- Sending events to CloudWatch Logs (p. 277)
- Creating CloudWatch alarms for CloudTrail events: examples (p. 282)
- Stopping CloudTrail from sending events to CloudWatch Logs (p. 287)
- CloudWatch log group and log stream naming for CloudTrail (p. 288)
- Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288)

Sending events to CloudWatch Logs

When you configure your trail to send events to CloudWatch Logs, CloudTrail sends only the events that match your trail settings. For example, if you configure your trail to log data events only, your trail sends data events only to your CloudWatch Logs log group. CloudTrail supports sending data, Insights, and management events to CloudWatch Logs. For more information, see Working with CloudTrail log files (p. 236).

To send events to a CloudWatch Logs log group:

- Make sure you have sufficient permissions to create or specify an IAM role. For more information, see Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console (p. 360).
- Create a new trail or specify an existing one. For more information, see Creating and updating a trail with the console (p. 78).
- Create a log group or specify an existing one.
- Specify an IAM role. If you are modifying an existing IAM role for an organization trail, you must manually update the policy to allow logging for the organization trail. For more information, see this policy example (p. 281) and Creating a trail for an organization (p. 125).
- Attach a role policy or use the default.

Contents

- Configuring CloudWatch Logs monitoring with the console (p. 278)
  - Creating a log group or specifying an existing log group (p. 278)
  - Specifying an IAM role (p. 279)
  - Viewing events in the CloudWatch console (p. 279)
- Configuring CloudWatch Logs monitoring with the Amazon CLI (p. 279)
  - Creating a log group (p. 279)
  - Creating a role (p. 280)
  - Creating a policy document (p. 280)
  - Updating the trail (p. 282)
- Limitation (p. 282)

Configuring CloudWatch Logs monitoring with the console

You can use the Amazon Web Services Management Console to configure your trail to send events to CloudWatch Logs for monitoring.

Creating a log group or specifying an existing log group

CloudTrail uses a CloudWatch Logs log group as a delivery endpoint for log events. You can create a log group or specify an existing one.

To create or specify a log group for an existing trail

1. Make sure you log in with an administrative user or role with sufficient permissions to configure CloudWatch Logs integration. For more information, see Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console (p. 360).
2. Open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
3. Choose the trail name. If you choose a trail that applies to all Regions, you will be redirected to the Region in which the trail was created. You can create a log group or choose an existing log group in the same Region as the trail.

   Note
   A trail that applies to all Regions sends log files from all Regions to the CloudWatch Logs log group that you specify.
4. In **CloudWatch Logs**, choose **Edit**.
5. For **CloudWatch Logs**, choose **Enabled**.
6. For **Log group name**, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name. For more information about naming, see CloudWatch log group and log stream naming for CloudTrail (p. 288).
7. If you choose **Existing**, choose a log group from the drop-down list.
8. For **Role name**, choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

   Note
   When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

9. Choose **Save changes**.

Specifying an IAM role

You can specify a role for CloudTrail to assume to deliver events to the log stream.

To specify a role

1. By default, the `CloudTrail_CloudWatchLogs_Role` is specified for you. The default role policy has the required permissions to create a CloudWatch Logs log stream in a log group that you specify, and to deliver CloudTrail events to that log stream.

   Note
   If you want to use this role for a log group for an organization trail, you must manually modify the policy after you create the role. For more information, see this policy example (p. 281) and Creating a trail for an organization (p. 125).

   a. To verify the role, go to the Amazon Identity and Access Management console at https://console.amazonaws.cn/iam/.
   b. Choose **Roles** and then choose the **CloudTrail_CloudWatchLogs_Role**.
   c. From the **Permissions** tab, expand the policy to view its contents.
2. You can specify another role, but you must attach the required role policy to the existing role if you want to use it to send events to CloudWatch Logs. For more information, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

Viewing events in the CloudWatch console

After you configure your trail to send events to your CloudWatch Logs log group, you can view the events in the CloudWatch console. CloudTrail typically delivers events to your log group within an average of about 5 minutes of an API call. This time is not guaranteed. Review the Amazon CloudTrail Service Level Agreement for more information.

To view events in the CloudWatch console

1. Open the CloudWatch console at https://console.amazonaws.cn/cloudwatch/.
2. In the left navigation pane, under **Logs**, choose **Log groups**.
3. Choose the log group that you specified for your trail.
4. Choose the log stream that you want to view.
5. To see the details of the event that your trail logged, choose an event.

   Note
   The **Time (UTC)** column in the CloudWatch console shows when the event was delivered to your log group. To see the actual time that the event was logged by CloudTrail, see the `eventTime` field.

Configuring CloudWatch Logs monitoring with the Amazon CLI

You can use the Amazon CLI to configure CloudTrail to send events to CloudWatch Logs for monitoring.

Creating a log group

1. If you don't have an existing log group, create a CloudWatch Logs log group as a delivery endpoint for log events using the CloudWatch Logs `create-log-group` command.

```
aws logs create-log-group --log-group-name name
```

The following example creates a log group named `CloudTrail/logs`:

```
aws logs create-log-group --log-group-name CloudTrail/logs
```

2. Retrieve the log group Amazon Resource Name (ARN).

```
aws logs describe-log-groups
```

Creating a role

Create a role for CloudTrail that enables it to send events to the CloudWatch Logs log group. The IAM `create-role` command takes two parameters: a role name and a file path to an assume role policy document in JSON format. The policy document that you use gives `AssumeRole` permissions to CloudTrail. The `create-role` command creates the role with the required permissions.

To create the JSON file that will contain the policy document, open a text editor and save the following policy contents in a file called `assume_role_policy_document.json`.

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

Run the following command to create the role with `AssumeRole` permissions for CloudTrail.

```
aws iam create-role --role-name role_name --assume-role-policy-document file://<path to
 assume_role_policy_document>.json
```

When the command completes, take a note of the role ARN in the output.

Creating a policy document

Create the following role policy document for CloudTrail. This document grants CloudTrail the permissions required to create a CloudWatch Logs log stream in the log group you specify and to deliver CloudTrail events to that log stream.

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

      "Sid": "AWSCloudTrailCreateLogStream2014110",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream"
```

```
      ],
      "Resource": [
        "arn:aws:logs:region:accountID:log-group:log_group_name:log-
stream:accountID_CloudTrail_region*"
      ]

    },
    {
      "Sid": "AWSCloudTrailPutLogEvents20141101",
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:region:accountID:log-group:log_group_name:log-
stream:accountID_CloudTrail_region*"
      ]
    }
  ]
}
```

Save the policy document in a file called `role-policy-document.json`.

If you're creating a policy that might be used for organization trails as well, you will need to configure it slightly differently. For example, the following policy grants CloudTrail the permissions required to create a CloudWatch Logs log stream in the log group you specify and to deliver CloudTrail events to that log stream for both trails in the Amazon account 111111111111 and for organization trails created in the 111111111111 account that are applied to the Amazon Organizations organization with the ID of *o-exampleorgid*:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailCreateLogStream20141101",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:o-exampleorgid_*"
            ]
        },
        {
            "Sid": "AWSCloudTrailPutLogEvents20141101",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:CloudTrail/
DefaultLogGroupTest:log-stream:o-exampleorgid_*"
            ]
        }
    ]
}
```

For more information about organization trails, see Creating a trail for an organization (p. 125).

Run the following command to apply the policy to the role.

```
aws iam put-role-policy --role-name role_name --policy-name cloudtrail-policy --policy-
document file://<path to role-policy-document>.json
```

Updating the trail

Update the trail with the log group and role information using the CloudTrail `update-trail` command.

```
aws cloudtrail update-trail --name trail_name --cloud-watch-logs-log-group-
arn log_group_arn --cloud-watch-logs-role-arn role_arn
```

For more information about the Amazon CLI commands, see the Amazon CloudTrail Command Line Reference.

Limitation

CloudWatch Logs and EventBridge each allow a maximum event size of 256 KB. Although most service events have a maximum size of 256 KB, some services still have events that are larger. CloudTrail does not send these events to CloudWatch Logs or EventBridge.

Starting with CloudTrail event version 1.05, events have a maximum size of 256 KB. This is to help prevent exploitation by malicious actors, and allow events to be consumed by other Amazon services, such as CloudWatch Logs and EventBridge.

Creating CloudWatch alarms for CloudTrail events: examples

This topic describes how to configure alarms for CloudTrail events, and includes examples.

Topics

- Prerequisites (p. 282)
- Create a metric filter and create an alarm (p. 283)
- Example security group configuration changes (p. 283)
- Example Amazon Web Services Management Console sign-in failures (p. 284)
- Example: IAM policy changes (p. 285)
- Configuring notifications for CloudWatch Logs alarms (p. 287)

Prerequisites

Before you can use the examples in this topic, you must:

- Create a trail with the console or CLI.
- Create a log group, which you can do as part of creating a trail. For more information about creating a trail, see Creating a trail (p. 78).
- Specify or create an IAM role that grants CloudTrail the permissions to create a CloudWatch Logs log stream in the log group that you specify and to deliver CloudTrail events to that log stream. The default `CloudTrail_CloudWatchLogs_Role` does this for you.

For more information, see Sending events to CloudWatch Logs (p. 277). Examples in this section are performed in the Amazon CloudWatch Logs console. For more information about how to create metric

filters and alarms, see Creating metrics from log events using filters and Using Amazon CloudWatch alarms in the *Amazon CloudWatch User Guide*.

Create a metric filter and create an alarm

To create an alarm, you must first create a metric filter, and then configure an alarm based on the filter. The procedures are shown for all examples. For more information about syntax for metric filters and patterns for CloudTrail log events, see the JSON-related sections of Filter and pattern syntax in the *Amazon CloudWatch Logs User Guide*.

Example security group configuration changes

Follow this procedure to create an Amazon CloudWatch alarm that is triggered when configuration changes occur on security groups.

Create a metric filter

1. Open the CloudWatch console at https://console.amazonaws.cn/cloudwatch/.
2. In the navigation pane, under **Logs**, choose **Log groups**.
3. In the list of log groups, choose the log group that you created for your trail.
4. From the **Metric filters** or **Actions** menu, choose **Create metric filter**.
5. On the **Define pattern** page, in **Create filter pattern**, enter the following for **Filter pattern**.

```
{ ($.eventName = AuthorizeSecurityGroupIngress) || ($.eventName =
 AuthorizeSecurityGroupEgress) || ($.eventName = RevokeSecurityGroupIngress) ||
 ($.eventName = RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) ||
 ($.eventName = DeleteSecurityGroup) }
```

6. In **Test pattern**, leave defaults. Choose **Next**.
7. On the **Assign metric** page, for **Filter name**, enter **`SecurityGroupEvents`**.
8. In **Metric details**, turn on **Create new**, and then enter **`CloudTrailMetrics`** for **Metric namespace**.
9. For **Metric name**, type **`SecurityGroupEventCount`**.
10. For **Metric value**, type **1**.
11. Leave **Default value** blank.
12. Choose **Next**.
13. On the **Review and create** page, review your choices. Choose **Create metric filter** to create the filter, or choose **Edit** to go back and change values.

Create an alarm

After you create the metric filter, the CloudWatch Logs log group details page for your CloudTrail trail log group opens. Follow this procedure to create an alarm.

1. On the **Metric filters** tab, find the metric filter you created in the section called “Create a metric filter” (p. 283). Fill the check box for the metric filter. In the **Metric filters** bar, choose **Create alarm**.
2. For **Specify metric and conditions**, enter the following.
   a. For **Graph**, the line is set at **1** based on other settings you make when you create your alarm.
   b. For **Metric name**, keep the current metric name, **`SecurityGroupEventCount`**.
   c. For **Statistic**, keep the default, **`Sum`**.
   d. For **Period**, keep the default, **`5 minutes`**.

e. In **Conditions**, for **Threshold type**, choose **Static**.

f. For **Whenever *`metric_name`* is**, choose **Greater/Equal**.

g. For the threshold value, enter **1**.

h. In **Additional configuration**, leave defaults. Choose **Next**.

3. On the **Configure actions** page, choose **Notification**, and then choose **In alarm**, which indicates that the action is taken when the threshold of 1 change event in 5 minutes is crossed, and **SecurityGroupEventCount** is in an alarm state.

   a. For **Send a notification to the following SNS topic**, choose **Create new topic**.

   b. Enter **`SecurityGroupChanges_CloudWatch_Alarms_Topic`** as the name for the new Amazon SNS topic.

   c. In **Email endpoints that will receive the notification**, enter the email addresses of users whom you want to receive notifications if this alarm is raised. Separate email addresses with commas.

      Each email recipient will receive an email asking them to confirm that they want to be subscribed to the Amazon SNS topic.

   d. Choose **Create topic**.

4. For this example, skip the other action types. Choose **Next**.

5. On the **Add name and description** page, enter a friendly name for the alarm, and a description. For this example, enter **`Security group configuration changes`** for the name, and **`Raises alarms if security group configuration changes occur`** for the description. Choose **Next**.

6. On the **Preview and create** page, review your choices. Choose **Edit** to make changes, or choose **Create alarm** to create the alarm.

   After you create the alarm, CloudWatch opens the **Alarms** page. The alarm's **Actions** column shows **Pending confirmation** until all email recipients on the SNS topic have confirmed that they want to subscribe to SNS notifications.

Example Amazon Web Services Management Console sign-in failures

Follow this procedure to create an Amazon CloudWatch alarm that is triggered when there are three or more Amazon Web Services Management Console sign-in failures during a five minute period.

Create a metric filter

1. Open the CloudWatch console at https://console.amazonaws.cn/cloudwatch/.
2. In the navigation pane, under **Logs**, choose **Log groups**.
3. In the list of log groups, choose the log group that you created for your trail.
4. From the **Metric filters** or **Actions** menu, choose **Create metric filter**.
5. On the **Define pattern** page, in **Create filter pattern**, enter the following for **Filter pattern**.

```
{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed authentication") }
```

6. In **Test pattern**, leave defaults. Choose **Next**.
7. On the **Assign metric** page, for **Filter name**, enter **`ConsoleSignInFailures`**.
8. In **Metric details**, turn on **Create new**, and then enter **`CloudTrailMetrics`** for **Metric namespace**.
9. For **Metric name**, type **`ConsoleSigninFailureCount`**.
10. For **Metric value**, type **1**.

11. Leave **Default value** blank.
12. Choose **Next**.
13. On the **Review and create** page, review your choices. Choose **Create metric filter** to create the filter, or choose **Edit** to go back and change values.

Create an alarm

After you create the metric filter, the CloudWatch Logs log group details page for your CloudTrail trail log group opens. Follow this procedure to create an alarm.

1. On the **Metric filters** tab, find the metric filter you created in the section called "Create a metric filter" (p. 284). Fill the check box for the metric filter. In the **Metric filters** bar, choose **Create alarm**.
2. On the **Create Alarm** page, in **Specify metric and conditions**, enter the following.
   a. For **Graph**, the line is set at **3** based on other settings you make when you create your alarm.
   b. For **Metric name**, keep the current metric name, **`ConsoleSigninFailureCount`**.
   c. For **Statistic**, keep the default, **`Sum`**.
   d. For **Period**, keep the default, **`5 minutes`**.
   e. In **Conditions**, for **Threshold type**, choose **Static**.
   f. For **Whenever *`metric_name`* is**, choose **Greater/Equal**.
   g. For the threshold value, enter **3**.
   h. In **Additional configuration**, leave defaults. Choose **Next**.
3. On the **Configure actions** page, for **Notification**, choose **In alarm**, which indicates that the action is taken when the threshold of 3 change events in 5 minutes is crossed, and **ConsoleSigninFailureCount** is in an alarm state.
   a. For **Send a notification to the following SNS topic**, choose **Create new topic**.
   b. Enter **`ConsoleSignInFailures_CloudWatch_Alarms_Topic`** as the name for the new Amazon SNS topic.
   c. In **Email endpoints that will receive the notification**, enter the email addresses of users whom you want to receive notifications if this alarm is raised. Separate email addresses with commas.

      Each email recipient will receive an email asking them to confirm that they want to be subscribed to the Amazon SNS topic.
   d. Choose **Create topic**.
4. For this example, skip the other action types. Choose **Next**.
5. On the **Add name and description** page, enter a friendly name for the alarm, and a description. For this example, enter **`Console sign-in failures`** for the name, and **`Raises alarms if more than 3 console sign-in failures occur in 5 minutes`** for the description. Choose **Next**.
6. On the **Preview and create** page, review your choices. Choose **Edit** to make changes, or choose **Create alarm** to create the alarm.

   After you create the alarm, CloudWatch opens the **Alarms** page. The alarm's **Actions** column shows **Pending confirmation** until all email recipients on the SNS topic have confirmed that they want to subscribe to SNS notifications.

Example: IAM policy changes

Follow this procedure to create an Amazon CloudWatch alarm that is triggered when an API call is made to change an IAM policy.

Create a metric filter

1. Open the CloudWatch console at https://console.amazonaws.cn/cloudwatch/.
2. In the navigation pane, choose **Logs**.
3. In the list of log groups, choose the log group that you created for your trail.
4. Choose **Actions**, and then choose **Create metric filter**.
5. On the **Define pattern** page, in **Create filter pattern**, enter the following for **Filter pattern**.

```
{($.eventName=DeleteGroupPolicy)||($.eventName=DeleteRolePolicy)||
($.eventName=DeleteUserPolicy)||($.eventName=PutGroupPolicy)||
($.eventName=PutRolePolicy)||($.eventName=PutUserPolicy)||($.eventName=CreatePolicy)||
($.eventName=DeletePolicy)||($.eventName=CreatePolicyVersion)||
($.eventName=DeletePolicyVersion)||($.eventName=AttachRolePolicy)||
($.eventName=DetachRolePolicy)||($.eventName=AttachUserPolicy)||
($.eventName=DetachUserPolicy)||($.eventName=AttachGroupPolicy)||
($.eventName=DetachGroupPolicy)}
```

6. In **Test pattern**, leave defaults. Choose **Next**.
7. On the **Assign metric** page, for **Filter name**, enter **`IAMPolicyChanges`**.
8. In **Metric details**, turn on **Create new**, and then enter **`CloudTrailMetrics`** for **Metric namespace**.
9. For **Metric name**, type **`IAMPolicyEventCount`**.
10. For **Metric value**, type **1**.
11. Leave **Default value** blank.
12. Choose **Next**.
13. On the **Review and create** page, review your choices. Choose **Create metric filter** to create the filter, or choose **Edit** to go back and change values.

Create an alarm

After you create the metric filter, the CloudWatch Logs log group details page for your CloudTrail trail log group opens. Follow this procedure to create an alarm.

1. On the **Metric filters** tab, find the metric filter you created in the section called "Create a metric filter" (p. 286). Fill the check box for the metric filter. In the **Metric filters** bar, choose **Create alarm**.
2. On the **Create Alarm** page, in **Specify metric and conditions**, enter the following.
   a. For **Graph**, the line is set at **1** based on other settings you make when you create your alarm.
   b. For **Metric name**, keep the current metric name, **`IAMPolicyEventCount`**.
   c. For **Statistic**, keep the default, **`Sum`**.
   d. For **Period**, keep the default, **`5 minutes`**.
   e. In **Conditions**, for **Threshold type**, choose **Static**.
   f. For **Whenever *`metric_name`* is**, choose **Greater/Equal**.
   g. For the threshold value, enter **1**.
   h. In **Additional configuration**, leave defaults. Choose **Next**.
   i.
3. On the **Configure actions** page, for **Notification**, choose **In alarm**, which indicates that the action is taken when the threshold of 1 change event in 5 minutes is crossed, and **IAMPolicyEventCount** is in an alarm state.
   a. For **Send a notification to the following SNS topic**, choose **Create new topic**.

b. Enter **`IAM_Policy_Changes_CloudWatch_Alarms_Topic`** as the name for the new Amazon SNS topic.

c. In **Email endpoints that will receive the notification**, enter the email addresses of users whom you want to receive notifications if this alarm is raised. Separate email addresses with commas.

   Each email recipient will receive an email asking them to confirm that they want to be subscribed to the Amazon SNS topic.

d. Choose **Create topic**.

4. For this example, skip the other action types. Choose **Next**.
5. On the **Add name and description** page, enter a friendly name for the alarm, and a description. For this example, enter **`IAM Policy Changes`** for the name, and **`Raises alarms if IAM policy changes occur`** for the description. Choose **Next**.
6. On the **Preview and create** page, review your choices. Choose **Edit** to make changes, or choose **Create alarm** to create the alarm.

   After you create the alarm, CloudWatch opens the **Alarms** page. The alarm's **Actions** column shows **Pending confirmation** until all email recipients on the SNS topic have confirmed that they want to subscribe to SNS notifications.

Configuring notifications for CloudWatch Logs alarms

You can configure CloudWatch Logs to send a notification whenever an alarm is triggered for CloudTrail. Doing so enables you to respond quickly to critical operational events captured in CloudTrail events and detected by CloudWatch Logs. CloudWatch uses Amazon Simple Notification Service (SNS) to send email. For more information, see Set Up Amazon SNS in the *CloudWatch Developer Guide*.

Stopping CloudTrail from sending events to CloudWatch Logs

You can stop sending Amazon CloudTrail events to Amazon CloudWatch Logs by updating a trail to disable CloudWatch Logs settings.

Stop sending events to CloudWatch Logs (console)

To stop sending CloudTrail events to CloudWatch Logs

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Trails**.
3. Choose the name of the trail for which you want to disable CloudWatch Logs integration.
4. In **CloudWatch Logs**, choose **Edit**.
5. Clear the **Enabled** check box.
6. Choose **Save changes**.

Stop sending events to CloudWatch Logs (CLI)

You can remove the CloudWatch Logs log group as a delivery endpoint by running the **update-trail** (p. 100) command. The following command clears the log group and role from the trail

configuration by replacing the values for the log group ARN and CloudWatch Logs role ARN with empty values.

```
aws cloudtrail update-trail --name trail_name --cloud-watch-logs-log-group-arn="" --cloud-watch-logs-role-arn=""
```

CloudWatch log group and log stream naming for CloudTrail

Amazon CloudWatch will display the log group that you created for CloudTrail events alongside any other log groups you have in a Region. We recommend that you use a log group name that helps you easily distinguish the log group from others. For example, **`CloudTrail/logs`**.

Follow these guidelines when naming a log group:

- Log group names must be unique within a Region for an Amazon Web Services account.
- Log group names can be between 1 and 512 characters long.
- Log group names consist of the following characters: a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), '/' (forward slash), '.' (period), and '#' (number sign).

When CloudTrail creates the log stream for the log group, it names the log stream according to the following format: *account_ID*_CloudTrail_*trail_region*.

> **Note**
> If the volume of CloudTrail logs is large, multiple log streams may be created to deliver log data to your log group. When there are multiple log streams, CloudTrail names each log stream according to the following format: *account_ID*_CloudTrail_*trail_region_number*.

For more information about CloudWatch log groups, see Working with log groups and log streams in the *Amazon CloudWatch Logs User Guide* and CreateLogGroup in the *Amazon CloudWatch Logs API Reference*.

Role policy document for CloudTrail to use CloudWatch Logs for monitoring

This section describes the permissions policy required for the CloudTrail role to send log events to CloudWatch Logs. You can attach a policy document to a role when you configure CloudTrail to send events, as described in Sending events to CloudWatch Logs (p. 277). You can also create a role using IAM. For more information, see Creating a Role for an Amazon Service (Amazon Web Services Management Console) or Creating a Role (CLI and API).

The following example policy document contains the permissions required to create a CloudWatch log stream in the log group that you specify and to deliver CloudTrail events to that log stream in the US East (Ohio) Region. (This is the default policy for the default IAM role `CloudTrail_CloudWatchLogs_Role`.)

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

      "Sid": "AWSCloudTrailCreateLogStream2014110",
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogStream"
      ],
      "Resource": [
        "arn:aws:logs:us-east-2:accountID:log-group:log_group_name:log-
stream:CloudTrail_log_stream_name_prefix*"
      ]

    },
    {
      "Sid": "AWSCloudTrailPutLogEvents20141101",
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-2:accountID:log-group:log_group_name:log-
stream:CloudTrail_log_stream_name_prefix*"
      ]
    }
  ]
}
```

If you're creating a policy that might be used for organization trails as well, you will need to modify it from the default policy created for the role. For example, the following policy grants CloudTrail the permissions required to create a CloudWatch Logs log stream in the log group you specify as the value of *log_group_name*, and to deliver CloudTrail events to that log stream for both trails in the Amazon account 111111111111 and for organization trails created in the 111111111111 account that are applied to the Amazon Organizations organization with the ID of *o-exampleorgid*:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailCreateLogStream20141101",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:log_group_name:log-
stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:log_group_name:log-stream:o-
exampleorgid_*"
            ]
        },
        {
            "Sid": "AWSCloudTrailPutLogEvents20141101",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-2:111111111111:log-group:log_group_name:log-
stream:111111111111_CloudTrail_us-east-2*",
                "arn:aws:logs:us-east-2:111111111111:log-group:log_group_name:log-stream:o-
exampleorgid_*"
            ]
        }
    ]
}
```

For more information about organization trails, see Creating a trail for an organization (p. 125).

Receiving CloudTrail log files from multiple accounts

You can have CloudTrail deliver log files from multiple Amazon accounts into a single Amazon S3 bucket. For example, you have four Amazon accounts with account IDs 111111111111, 222222222222, 333333333333, and 444444444444, and you want to configure CloudTrail to deliver log files from all four of these accounts to a bucket belonging to account 111111111111. To accomplish this, complete the following steps in order:

1. Turn on CloudTrail in the account where the destination bucket will belong (111111111111 in this example). Do not turn on CloudTrail in any other accounts yet.

   For instructions, see Creating a trail (p. 78).
2. Update the bucket policy on your destination bucket to grant cross-account permissions to CloudTrail.

   For instructions, see Setting bucket policy for multiple accounts (p. 291).
3. Turn on CloudTrail in the other accounts you want (222222222222, 333333333333, and 444444444444 in this example). Configure CloudTrail in these accounts to use the same bucket belonging to the account that you specified in step 1 (111111111111 in this example).

   For instructions, see Turning on CloudTrail in additional accounts (p. 292).

Redacting bucket owner account IDs for data events called by other accounts

Historically, if CloudTrail data events were enabled in the Amazon account of an Amazon S3 data event API caller, CloudTrail showed the account ID of the S3 bucket owner in the data event (such as `PutObject`). This occurred even if the bucket owner account did not have S3 data events enabled.

Now, CloudTrail removes the account ID of the S3 bucket owner in the `resources` block if both of the following conditions are met:

- The data event API call is from a different Amazon account than the Amazon S3 bucket owner.
- The API caller received an `AccessDenied` error that was only for the caller account.

The owner of the resource on which the API call was made still receives the full event.

The following event record snippets are an example of the expected behavior. In the `Historic` snippet, the account ID 123456789012 of the S3 bucket owner is shown to an API caller from a different account. In the example of current behavior, the account ID of the bucket owner is not shown.

```
# Historic

"resources": [
    {
        "type": "AWS::S3::Object",
        "ARNPrefix": "arn:aws:s3:::test-my-bucket-2/"
    },
    {
        "accountId": "123456789012",
        "type": "AWS::S3::Bucket",
        "ARN": "arn:aws:s3:::test-my-bucket-2"
```

```
    }
]
```

The following is the current behavior.

```
# Current

"resources": [
    {
        "type": "AWS::S3::Object",
        "ARNPrefix": "arn:aws:s3:::test-my-bucket-2/"
    },
    {
        "accountId": "",
        "type": "AWS::S3::Bucket",
        "ARN": "arn:aws:s3:::test-my-bucket-2"
    }
]
```

Topics

- Setting bucket policy for multiple accounts (p. 291)
- Turning on CloudTrail in additional accounts (p. 292)

Setting bucket policy for multiple accounts

For a bucket to receive log files from multiple accounts, its bucket policy must grant CloudTrail permission to write log files from all the accounts you specify. This means that you must modify the bucket policy on your destination bucket to grant CloudTrail permission to write log files from each specified account.

> **Note**
> For security reasons, unauthorized users cannot create a trail that includes `AWSLogs/` as the `S3KeyPrefix` parameter.

To modify bucket permissions so that files can be received from multiple accounts

1. Sign in to the Amazon Web Services Management Console using the account that owns the bucket (111111111111 in this example) and open the Amazon S3 console.
2. Choose the bucket where CloudTrail delivers your log files and then choose **Permissions**.
3. For **Bucket policy**, choose **Edit**.
4. Modify the existing policy to add a line for each additional account whose log files you want delivered to this bucket. See the following example policy and note the underlined `Resource` line specifying a second account ID. As a security best practice, add an `aws:SourceArn` condition key to the Amazon S3 bucket policy. This helps prevent unauthorized access to your S3 bucket. If you have existing trails, be sure to add one or more condition keys.

   > **Note**
   > An Amazon account ID is a twelve-digit number, including leading zeros.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck20131101",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
```

```
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws-cn:s3:::myBucketName",
      "Condition": {
          "StringEquals": {
            "aws:SourceArn": [
              "arn:aws:cloudtrail:region:111111111111:trail/primaryTrailName",
              "arn:aws:cloudtrail:region:222222222222:trail/secondaryTrailName"
            ]
          }
       }
    },
    {
      "Sid": "AWSCloudTrailWrite20131101",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws-cn:s3:::myBucketName/optionalLogFilePrefix/AWSLogs/111111111111/*",
        "arn:aws-cn:s3:::myBucketName/optionalLogFilePrefix/AWSLogs/222222222222/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:cloudtrail:region:111111111111:trail/primaryTrailName",
            "arn:aws:cloudtrail:region:222222222222:trail/secondaryTrailName"
          ],
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

Turning on CloudTrail in additional accounts

You can use the console or the command line interface to turn on CloudTrail in additional Amazon accounts.

Using the console to turn on CloudTrail in additional Amazon accounts

You can use the CloudTrail console to turn on CloudTrail in additional accounts.

1. Sign into the Amazon management console using account 222222222222 credentials and open the Amazon CloudTrail console. In the navigation bar, select the Region where you want to turn on CloudTrail.
2. On the CloudTrail service home page, the **Trails** page, or the **Trails** section of the **Dashboard** page, choose **Create trail**.
3. On the **Create Trail** page, for **Trail name**, type a name for your trail. For more information, see CloudTrail naming requirements (p. 151).
4. If this is an Amazon Organizations organization trail, you can choose to enable the trail for all accounts in your organization. You only see this option if you sign in to the console with a user or role in the management account or a delegated administrator account. To create an organization trail, the user or role must have sufficient permissions (p. 128). For more information, see Creating a trail for an organization (p. 125).

5. For **Storage location**, choose **Use existing S3 bucket**. Use the text box to enter the name of the bucket you created previously for storing log files when you signed in using account 111111111111 credentials.

   Note
   The bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 bucket policy for CloudTrail (p. 363).

   To make it easier to find your logs, create a new folder (also known as a *prefix*) in an existing bucket to store your CloudTrail logs. Enter the prefix in **Prefix**.

6. For **Prefix**, enter the same prefix you entered for storing log files when you turned on CloudTrail using account 111111111111 credentials. If you choose to use a prefix that is different from the one you entered when you turned on CloudTrail in the first account, you must edit the bucket policy on your destination bucket to allow CloudTrail to write log files to your bucket using this new prefix.

7. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

   If you enable SSE-KMS encryption, choose a **New** or **Existing** Amazon KMS key. In **Amazon KMS Alias**, specify an alias, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394). CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

   Note
   You can also type the ARN of a key from another account. The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).

8. In **Additional settings**, configure the following.

   a. For **Log file validation**, choose **Enabled** to have log digests delivered to your S3 bucket. You can use the digest files to verify that your log files did not change after CloudTrail delivered them. For more information, see Validating CloudTrail log file integrity (p. 307).

   b. For **SNS notification delivery**, choose **Enabled** to be notified each time a log is delivered to your bucket. CloudTrail stores multiple events in a log file. SNS notifications are sent for every log file, not for every event. For more information, see Configuring Amazon SNS notifications for CloudTrail (p. 147).

      If you enable SNS notifications, for **Create a new SNS topic**, choose **New** to create a topic, or choose **Existing** to use an existing topic. If you are creating a trail that applies to all Regions, SNS notifications for log file deliveries from all Regions are sent to the single SNS topic that you create.

      If you choose **New**, CloudTrail specifies a name for the new topic for you, or you can type a name. If you choose **Existing**, choose an SNS topic from the drop-down list. You can also enter the ARN of a topic from another Region or from an account with appropriate permissions. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

      If you create a topic, you must subscribe to the topic to be notified of log file delivery. You can subscribe from the Amazon SNS console. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. For more information, see the Amazon Simple Notification Service Getting Started Guide.

9. Optionally, configure CloudTrail to send log files to CloudWatch Logs by choosing **Enabled** in **CloudWatch Logs**. For more information, see Sending events to CloudWatch Logs (p. 277).

    a. If you enable integration with CloudWatch Logs, choose **New** to create a new log group, or **Existing** to use an existing one. If you choose **New**, CloudTrail specifies a name for the new log group for you, or you can type a name.

    b. If you choose **Existing**, choose a log group from the drop-down list.

    c. Choose **New** to create a new IAM role for permissions to send logs to CloudWatch Logs. Choose **Existing** to choose an existing IAM role from the drop-down list. The policy statement for the new or existing role is displayed when you expand **Policy document**. For more information about this role, see Role policy document for CloudTrail to use CloudWatch Logs for monitoring (p. 288).

    Note
    When you configure a trail, you can choose an S3 bucket and SNS topic that belong to another account. However, if you want CloudTrail to deliver events to a CloudWatch Logs log group, you must choose a log group that exists in your current account.

10. For **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups and Why use tags for trails? (p. 11).

11. On the **Choose log events** page, choose the event types that you want to log. For **Management events**, do the following.

    a. For **API activity**, choose if you want your trail to log **Read** events, **Write** events, or both. For more information, see Management events (p. 238).

    b. Choose **Exclude Amazon KMS events** to filter Amazon Key Management Service (Amazon KMS) events out of your trail. The default setting is to include all Amazon KMS events.

    The option to log or exclude Amazon KMS events is available only if you log management events on your trail. If you choose not to log management events, Amazon KMS events are not logged, and you cannot change Amazon KMS event logging settings.

    Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` typically generate a large volume (more than 99%) of events. These actions are now logged as **Read** events. Low-volume, relevant Amazon KMS actions such as `Disable`, `Delete`, and `ScheduleKey` (which typically account for less than 0.5% of Amazon KMS event volume) are logged as **Write** events.

    To exclude high-volume events like `Encrypt`, `Decrypt`, and `GenerateDataKey`, but still log relevant events such as `Disable`, `Delete` and `ScheduleKey`, choose to log **Write** management events, and clear the check box for **Exclude Amazon KMS events**.

12. For **Data events**, you can specify logging data events for Amazon S3 buckets, Amazon Lambda functions, Amazon DynamoDB tables, or a combination of these resource types. By default, trails don't log data events. Additional charges apply for logging data events. For more information, see Data events (p. 243). For CloudTrail pricing, see Amazon CloudTrail Pricing. More data event types are available if you use advanced event selectors; for more information, see Creating a trail in the console (advanced event selectors) (p. 82) in this topic.

    For Amazon S3 buckets:

    a. For **Data event source**, choose **S3**.

    b. You can choose to log **All current and future S3 buckets**, or you can specify individual buckets or functions. By default, data events are logged for all current and future S3 buckets.

    Note
    Keeping the default **All current and future S3 buckets** option enables data event logging for all buckets currently in your Amazon account and any buckets you create

after you finish creating the trail. It also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a bucket that belongs to another Amazon account.
If you are creating a trail for a single Region (done by using the Amazon CLI), choosing **All current and future S3 buckets** enables data event logging for all buckets in the same Region as your trail and any buckets you create later in that Region. It will not log data events for Amazon S3 buckets in other Regions in your Amazon account.

c. If you leave the default, **All current and future S3 buckets**, choose to log **Read** events, **Write** events, or both.

d. To select individual buckets, empty the **Read** and **Write** check boxes for **All current and future S3 buckets**. In **Individual bucket selection**, browse for a bucket on which to log data events. Find specific buckets by typing a bucket prefix for the bucket you want. You can select multiple buckets in this window. Choose **Add bucket** to log data events for more buckets. Choose to log **Read** events, such as `GetObject`, **Write** events, such as `PutObject`, or both.

This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.

To remove a bucket from logging, choose **X**.

13. To add another data type on which to log data events, choose **Add data event type**.
14. For Lambda functions:

a. For **Data event source**, choose **Lambda**.

b. In **Lambda function**, choose **All regions** to log all Lambda functions, or **Input function as ARN** to log data events on a specific function.

To log data events for all Lambda functions in your Amazon account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.

Note
If you are creating a trail for all Regions, this selection enables data event logging for all functions currently in your Amazon account, and any Lambda functions you might create in any Region after you finish creating the trail. If you are creating a trail for a single Region (done by using the Amazon CLI), this selection enables data event logging for all functions currently in that Region in your Amazon account, and any Lambda functions you might create in that Region after you finish creating the trail. It does not enable data event logging for Lambda functions created in other Regions.
Logging data events for all functions also enables logging of data event activity performed by any IAM identity in your Amazon account, even if that activity is performed on a function that belongs to another Amazon account.

c. If you choose **Input function as ARN**, enter the ARN of a Lambda function.

Note
If you have more than 15,000 Lambda functions in your account, you cannot view or select all functions in the CloudTrail console when creating a trail. You can still select the option to log all functions, even if they are not displayed. If you want to log data events for specific functions, you can manually add a function if you know its ARN. You can also finish creating the trail in the console, and then use the Amazon CLI and the **put-event-selectors** command to configure data event logging for specific Lambda functions. For more information, see Managing trails with the Amazon CLI (p. 103).

15. For DynamoDB tables:

a. For **Data event source**, choose **DynamoDB**.

b. In **DynamoDB table selection**, choose **Browse** to select a table, or paste in the ARN of a DynamoDB table to which you have access. A DynamoDB table ARN uses the following format:

```
arn:partition:dynamodb:region:account_ID:table/table_name
```

To add another table, choose **Add row**, and browse for a table or paste in the ARN of a table to which you have access.

16. Choose **Insights events** if you want your trail to log CloudTrail Insights events.

    In **Event type**, select **Insights events**. In **Insights events**, choose **API call rate**, **API error rate**, or both. You must be logging **Write** management events to log Insights events for **API call rate**. You must be logging **Read** or **Write** management events to log Insights events for **API error rate**.

    CloudTrail Insights analyzes management events for unusual activity, and logs events when anomalies are detected. By default, trails don't log Insights events. For more information about Insights events, see Logging Insights events for trails (p. 267). Additional charges apply for logging Insights events. For CloudTrail pricing, see Amazon CloudTrail Pricing.

    Insights events are delivered to a different folder named `/CloudTrail-Insightof` the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named `S3bucketName/AWSLogs/CloudTrail/`, the S3 bucket name with a new prefix is named `S3bucketName/AWSLogs/CloudTrail-Insight/`.

17. When you are finished choosing event types to log, choose **Next**.
18. On the **Review and create** page, review your choices. Choose **Edit** in a section to change the trail settings shown in that section. When you are ready to create the trail, choose **Create trail**.
19. The new trail appears on the **Trails** page. The **Trails** page shows the trails in your account from all Regions. In about 5 minutes, CloudTrail publishes log files that show the Amazon API calls made in your account. You can see the log files in the S3 bucket that you specified. It can take up to 36 hours for CloudTrail to deliver the first Insights event, if you have enabled Insights event logging, and unusual activity is detected.

CloudTrail starts publishing log files that show the Amazon calls made in your accounts in this Region. CloudTrail typically delivers logs within an average of about 5 minutes of an API call. This time is not guaranteed.

Using the CLI to turn on CloudTrail in additional Amazon accounts

You can use the Amazon command line tools to turn on CloudTrail in additional accounts and aggregate their log files to one Amazon S3 bucket. For more information about these tools, see the Amazon Command Line Interface User Guide.

Turn on CloudTrail in your additional accounts by using the `create-trail` command, specifying the following:

- `--name` specifies the name of the trail.
- `--s3-bucket-name` specifies the existing Amazon S3 bucket you created when you turned on CloudTrail in your first account (111111111111 in this example).
- `--s3-prefix` specifies a prefix for the log file delivery path (optional).
- `--is-multi-region-trail` specifies that this trail will log events in all Amazon Regions.

In contrast to trails that you create using the console, you must give every trail you create with the Amazon CLI a name. You can create one trail for each Region in which an account is running Amazon resources.

The following example command shows how to create a trail for your additional accounts by using the Amazon CLI. To have log files for these account delivered to the bucket you created in your first account (111111111111 in this example), specify the bucket name in the `--s3-bucket-name` option. Amazon S3 bucket names are globally unique.

```
aws cloudtrail create-trail --name my-trail --s3-bucket-name my-bucket --is-multi-region-
trail
```

When you run the command, you will see output similar to the following:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "AWSCloudTrailExample",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:222222222222:trail/my-trail",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "IsOrganizationTrail": false,
    "S3BucketName": "MyBucketBelongingToAccount111111111111"
}
```

For more information about using CloudTrail from the Amazon command line tools, see the CloudTrail command line reference.

Sharing CloudTrail log files between Amazon accounts

This section explains how to share CloudTrail log files between multiple Amazon accounts. We will assume that the log files have all been received in a single Amazon S3 bucket, which is the default setting for a trail created in the CloudTrail console. In the first scenario, you will learn how to grant read-only access to the accounts that generated the log files that have been placed into your Amazon S3 bucket. In the second scenario, you will learn how to grant access to all of the log files to a third-party account that can analyze the files for you.

To share log files between multiple Amazon accounts, you must perform the following general steps. These steps are explained in detail later in this section.

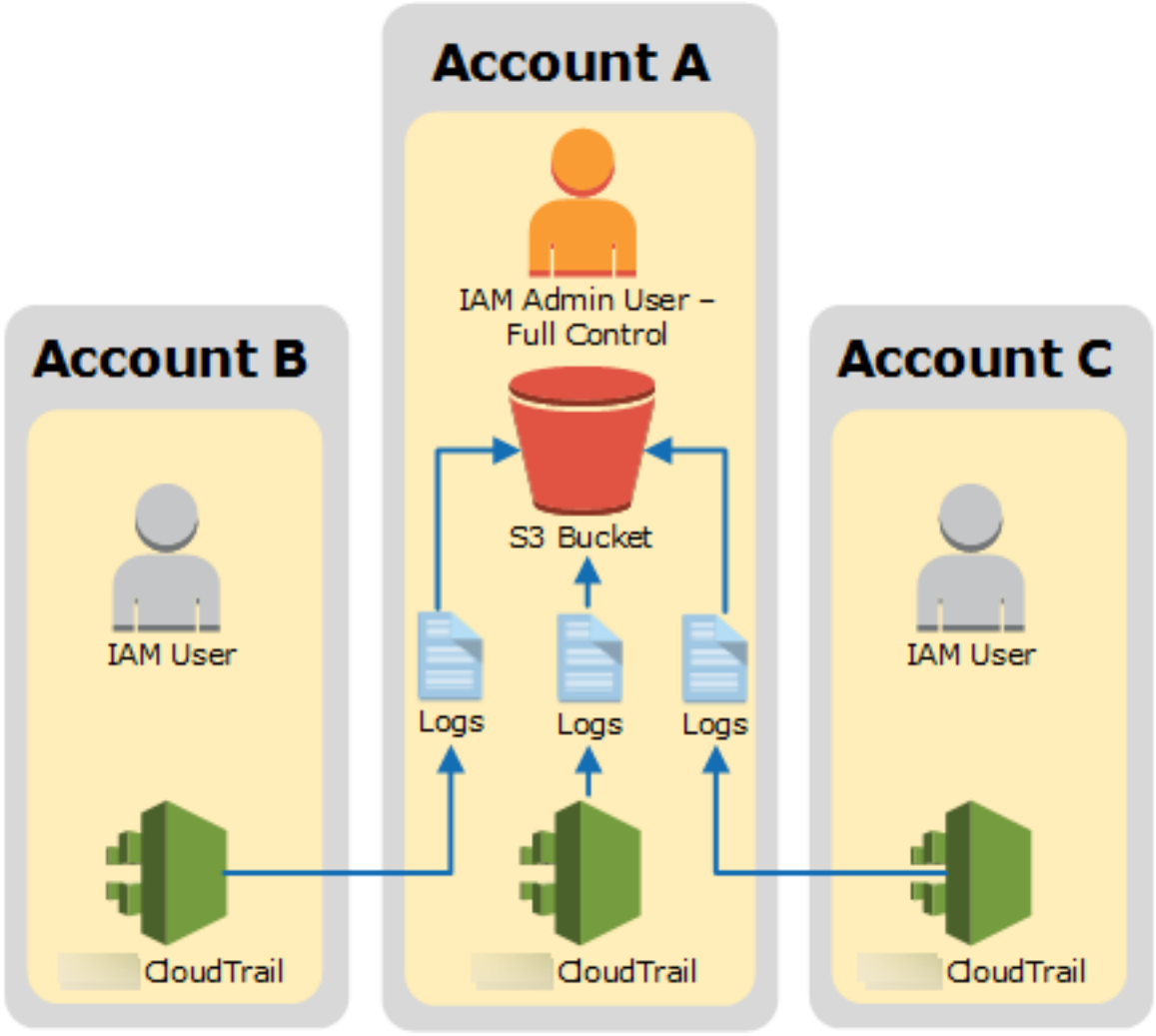
- Create an IAM role for each account that you want to share log files with.
- For each of these IAM roles, create an access policy that grants read-only access to the account you want to share the log files with.
- Have a user in each account programmatically assume the appropriate role and retrieve the log files.

This section walks you through the preceding steps in the context of two different sharing scenarios: granting access to the log files to each account that generated those files, and sharing log files with a third party. Most of the steps you take for the two scenarios are the same; the important difference is in what kind of permissions the IAM role grants to each account. That is, you can grant permission for an account to read only its own log files, or you can grant an account permission to read all log files. For details about permissions management for IAM roles, see IAM roles in *IAM User Guide*.

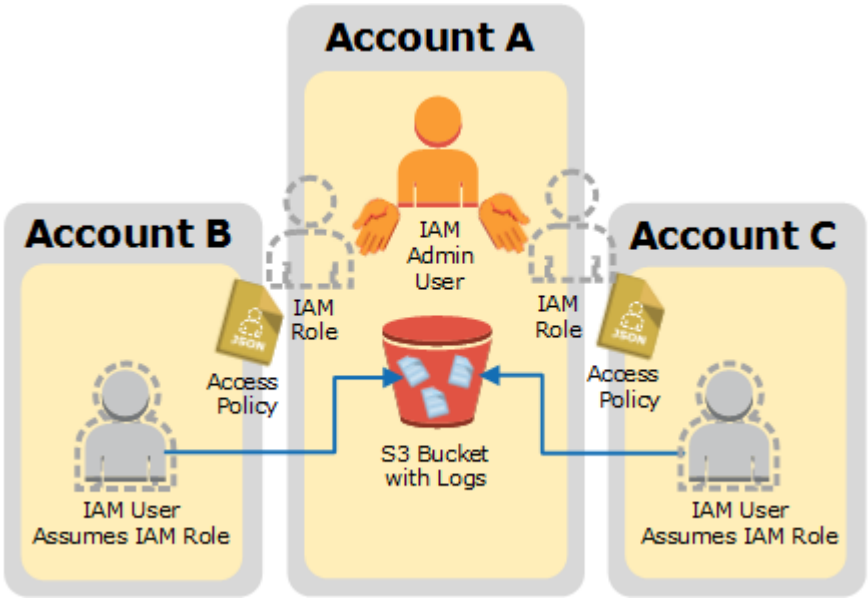
Scenario 1: Granting access to the account that generated the log files

In this scenario, we'll assume that your enterprise is made up of two business units and that it maintains three Amazon accounts. The first account, Account A, is the top-level account. For example, it might be managed by your enterprise's IT department and therefore be responsible for collecting log files from all other departments and business units into a single bucket. The other two accounts, B and C, correspond to your enterprise's business units.

This scenario assumes that you have already configured the log files from all three accounts to be delivered to a single Amazon S3 bucket, and that account A has full control over that bucket, as shown in the following illustration.



Although the Amazon S3 bucket contains log files that were generated by Accounts A, B and C, accounts B and C do not initially have access to the log files that accounts B and C generated. You will give each business unit read-only access to the log files that it generated, as shown in the following illustration.



To grant read-only access to the log files generated by accounts B and C, you must do the following in the account Account A. Remember that Account A has full control of the Amazon S3 bucket.

- Create an IAM role for account B and another IAM role for account C. For details, see Creating a role (p. 300)
- For the IAM role created for account B, create an access policy that grants read-only access to the log files generated by account B. For the IAM role created for account C, create an access policy that grants read-only access to the log files generated by account C. For details, see: Creating an access policy to grant access to accounts you own (p. 302)
- Have an IAM user in account B programmatically assume the role created for account B. Have an IAM user in account C programmatically assume the role created for account C. Each IAM user must be given permission to assume the role by the respective account owner. For details, see: Creating permissions policies for IAM users (p. 305).
- Finally, the account owner who grants the permission must be an administrator, and must know the ARN of the role in account A that is being assumed. For details, see: Calling AssumeRole (p. 306).

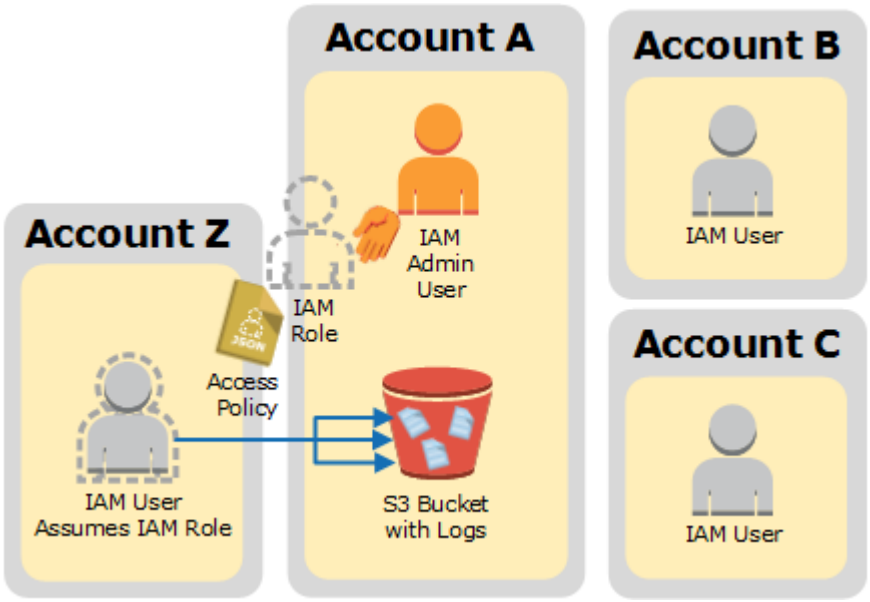
The users in accounts B and C can then programmatically retrieve their own log files, but not the log files of any other account.

Scenario 2: Granting access to all logs

In this scenario, we'll assume that your enterprise is structured as it was in the previous scenario, that is, it is made up of two business units and it maintains three Amazon accounts. The first account, Account A, is the top-level account. For example, it might be managed by your enterprise's IT department and therefore be responsible for placing all other log files into a single bucket. The other two accounts, B and C, correspond to each of your enterprise's business units.

Like the previous scenario, this scenario assumes that you have already placed the log files from all three accounts into a single Amazon S3 bucket, and that account A has full control over that bucket.

Finally, we'll also assume that your enterprise wants to share all the log files from all accounts (A, B, and C) with a third party. We'll say that the third party has an Amazon account called Account Z, as shown in the following illustration.



To share all of the log files from your enterprise with Account Z, you must do the following in the Account A, the account that has full control over the Amazon S3 bucket.

- Create an IAM role for Account Z. How: Creating a role (p. 300)
- For the IAM role created for Account Z, create an access policy that grants read-only access to the log files generated by accounts A, B, and C. How: Creating an access policy to grant access to a third party (p. 303)
- Have a user in Account Z programmatically assume the role and then retrieve the appropriate log files. The user must be given permission to assume the role by the owner of Account Z. How: Creating permissions policies for IAM users (p. 305). Further, the account owner who grants the permission must be an administrator and know the ARN of the role in Account A that is being assumed. How: Calling AssumeRole (p. 306).

Creating a role

When you aggregate log files from multiple accounts into a single Amazon S3 bucket, only the account that has full control of the bucket, Account A in our example, has full read access to all of the log files in the bucket. Accounts B, C, and Z in our example do not have any rights until granted. Therefore, to share your Amazon CloudTrail log files from one account to another (that is, to complete either Scenario 1 or Scenario 2 described previously in this section), you must *enable cross-account access*. You can do this by creating IAM roles and their associated access policies.

Roles

Create an IAM *role* for each account to which you want to give access. In our example, you will have three roles, one each for accounts B, C, and Z. Each IAM role defines an access or permissions policy that enables the accounts to access the resources (log files) owned by account A. The permissions are attached to each role and are associated with each account (B, C, or Z) only when the role is assumed. For details about permissions management for IAM roles, see IAM Roles in the *IAM User Guide*. For more information about how to assume a role, see Assuming a role (p. 304).

Policies

There are two policies for each IAM role you create. The *trust policy* specifies a *trusted entity* or *principal*. In our example, accounts B, C, and Z are trusted entities, and a user with the proper permissions in those accounts can assume the role.

You automatically create the *trust policy* when you use the console to create the role. If you use the SDK to create the role, you must supply the trust policy as a parameter to the `CreateRole` API. If you use the Amazon CLI to create the role, you must specify the trust policy in the `create-role` Amazon CLI command.

The *role access (or permissions) policy* that you must create as the owner of Account A defines what actions and resources the principal or trusted entity is allowed access to (in this case, the CloudTrail log files). For Scenario 1, this grants log file access to the account that generated the log files, as discussed in Creating an access policy to grant access to accounts you own (p. 302). For Scenario 2, this grants read access to all log files to a third party, as discussed in Creating an access policy to grant access to a third party (p. 303).

For further details about creating and working with IAM policies, see Access Management in the *IAM User Guide*.

Creating a role

To create a role by using the console

1. Sign into the Amazon Management Console as an administrator of Account A.
2. Navigate to the IAM console.
3. In the navigation pane, choose **Roles**.
4. Choose **Create role**.
5. For **Trusted entity type**, choose **Amazon Web Services account**.
6. Choose **Another Amazon Web Services account.**
7. For Scenario 1, do the following to provide access between accounts you own:
   a. Enter the twelve-digit account ID of the account (B, C, or Z) to be granted access.
   b. Check the **Require MFA** box if you want the user to provide multi-factor authentication before assuming the role.

   For Scenario 2, do the following to provide access to a third-party account. In our example, you would perform these steps for Account Z, the third-party log analyzer:
   a. Enter the twelve-digit account ID of the account (Account Z) to be granted access.
   b. Enter an external ID that provides additional control over who can assume the role. For more information, see How to Use an External ID When Granting Access to Your Amazon Resources to a Third Party in the *IAM User Guide*.
8. Choose **Next** to attach a policy that sets the permissions for this role.
9. Under **Permissions policies**, choose the **AmazonS3ReadOnlyAccess** policy.

   Note
   By default, the **AmazonS3ReadOnlyAccess** policy grants retrieval and list rights to all Amazon S3 buckets within your account.

   - To grant an account access to only that account's log files (Scenario 1), see Creating an access policy to grant access to accounts you own (p. 302).
   - To grant an account access to all of the log files in the Amazon S3 bucket (Scenario 2), see Creating an access policy to grant access to a third party (p. 303).
10. Choose **Next** to provide a role name and description.

11. Type a name and description for the new role, and then choose **Next**.
12. Choose **Create role**. When the role creation process completes, the role you created appears in the role list.

Creating an access policy to grant access to accounts you own

In Scenario 1, as an administrative user in Account A, you have full control over the Amazon S3 bucket to which CloudTrail writes log files for accounts B and C. You want to share each business unit's log files back to business unit that created them. But, you don't want a unit to be able to read any other unit's log files.

For example, to share Account B's log files with Account B but not with Account C, you must create a new IAM role in Account A that specifies that Account B is a trusted account. This role trust policy specifies that Account B is trusted to assume the role created by Account A, and should look like the following example. The trust policy is automatically created if you create the role by using the console. If you use the SDK to create the role, you must supply the trust policy as a parameter to the `CreateRole` API. If you use the CLI to create the role, you must specify the trust policy in the `create-role` CLI command.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-B-id:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

You must also create an access policy to specify that Account B can read from only the location to which B wrote its log files. The access policy will look something like the following. Note that the **Resource** ARN includes the twelve-digit account ID for Account B, and the prefix you specified, if any, when you turned on CloudTrail for Account B during the aggregation process. For more information about specifying a prefix, see Turning on CloudTrail in additional accounts (p. 292).

Important
You must ensure that the prefix in the access policy is exactly the same as the prefix that you specified when you turned on CloudTrail for Account B. If it is not, then you must edit the IAM role access policy in Account A to incorporate the actual prefix for Account B. If the prefix in the role access policy is not exactly the same as the prefix you specified when you turned on CloudTrail in Account B, then Account B will not be able to access its log files.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
```

```
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/account-B-id/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    }
  ]
}
```

The role you create for Account C will be nearly identical to the one you created for Account B. The access policy for each role must include the appropriate account ID and prefix so that each account can read from only the location to which CloudTrail wrote that account's log files.

After you create roles for each account and specify the appropriate trust and access policies, and after an IAM user in each account has been granted access by the administrator of that account, an IAM user in accounts B or C can programmatically assume the role.

After you create roles for each account and specify the appropriate trust and access policies, an IAM user in one of the newly trusted accounts (B or C) must programmatically assume the role in order to read log files from the Amazon S3 bucket.

For more information, see Assuming a role (p. 304).

Creating an access policy to grant access to a third party

Account A must create a separate IAM role for Account Z, the third-party analyzer in Scenario 2. When you create the role, Amazon automatically creates the trust relationship, which specifies that Account Z will be trusted to assume the role. The access policy for the role specifies what actions Account Z can take. For more information about creating roles and role policies, see Creating a role (p. 300).

For example, the trust relationship created by Amazon specifies that Account Z is trusted to assume the role created by Account A. The following is an example trust policy:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::account-Z-id:root"},
        "Action": "sts:AssumeRole"
    }]
}
```

If you specified an external ID when you created the role for Account Z, your access policy contains an added `Condition` element that tests the unique ID assigned by Account Z. The test is performed when the role is assumed. The following example access policy has a `Condition` element.

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

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::account-Z-id:root"},
        "Action": "sts:AssumeRole",
        "Condition": {"StringEquals": {"sts:ExternalId": "external-ID-issued-by-account-
Z"}}
    }]
}
```

You must also create an access policy for the Account A role to specify that Account Z can read all logs from the Amazon S3 bucket. The access policy should look something like the following example. The wild card (*) at the end of the `Resource` value indicates that Account Z can access any log file in the S3 bucket to which it has been granted access.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": "arn:aws:s3:::bucket-name/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

After you create a role for Account Z and specify the appropriate trust relationship and access policy, an IAM user in Account Z must programmatically assume the role to be able to read log files from the bucket. For more information, see Assuming a role (p. 304).

Assuming a role

You must designate a separate IAM user to assume each role you create in each account. You must then ensure that each IAM user has appropriate permissions.

IAM users and roles

After you create the necessary roles and policies in Account A for scenarios 1 and 2, you must designate an IAM user in each of the accounts B, C, and Z. Each IAM user programmatically assumes the appropriate role to access the log files. That is, the user in account B assumes the role created for account B, the user in account C assumes the role created for account C, and the user in account Z assumes the role created for account Z. When a user assumes a role, Amazon returns temporary security credentials to that user. They can then make requests to list, retrieve, copy, or delete log files depending on the permissions granted by the access policy associated with the role.

For more information about working with IAM identities, see IAM Identities (users, user groups, and roles).

The primary difference between scenarios 1 and 2 is in the access policy that you create for each IAM role in each scenario.

- In scenario 1, the access policies for accounts B and C limit each account to reading only its own log files. For more information, see Creating an access policy to grant access to accounts you own (p. 302).
- In scenario 2, the access policy for Account Z allows it to read all the log files that are aggregated in the Amazon S3 bucket. For more information, see Creating an access policy to grant access to a third party (p. 303).

Creating permissions policies for IAM users

To perform the actions permitted by a role, the IAM user must have permission to call the Amazon STS `AssumeRole` API. You must edit the policy for each user to grant them the appropriate permissions. To do this, you set a **Resource** element in the policy that you attach to the IAM user. The following example shows a policy for an IAM user in Account B that allows the user to assume a role named `Test` created earlier by Account A.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::account-A-id:role/Test"
    }
  ]
}
```

To edit a customer managed policy (console)

1. Sign in to the Amazon Web Services Management Console and open the IAM console at https://console.amazonaws.cn/iam/.
2. In the navigation pane, choose **Policies**.
3. In the list of policies, choose the policy name of the policy to edit. You can use the search box to filter the list of policies.
4. Choose the **Permissions** tab, and then choose **Edit**.
5. Do one of the following:

   - Choose the **Visual** option to change your policy without understanding JSON syntax. You can make changes to the service, actions, resources, or optional conditions for each permission block in your policy. You can also import a policy to add additional permissions to the bottom of your policy. When you are finished making changes, choose **Next** to continue.
   - Choose the **JSON** option to modify your policy by typing or pasting text in the JSON text box. You can also import a policy to add additional permissions to the bottom of your policy. Resolve any security warnings, errors, or general warnings generated during policy validation, and then choose **Next**.

     Note
     You can switch between the **Visual** and **JSON** editor options any time. However, if you make changes or choose **Next** in the **Visual** editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the *IAM User Guide*.
6. On the **Review and save** page, review **Permissions defined in this policy** and then choose **Save changes** to save your work.

7. If the managed policy already has the maximum of five versions, choosing **Save changes** displays a dialog box. To save your new version, the oldest non-default version of the policy is removed and replaced with this new version. Optionally, you can set the new version as the default policy version.

   Choose **Save changes** to save your new policy version.

Calling AssumeRole

A user in accounts B, C, or Z can assume a role by creating an application that calls the Amazon STS `AssumeRole` API and passes the role session name, the Amazon Resource Number (ARN) of the role to assume, and an optional external ID. The role session name is defined by Account A when it creates the role to assume. The external ID, if any, is defined by Account Z and passed to Account A for inclusion during role creation. For more information, see How to Use an External ID When Granting Access to Your Amazon Resources to a Third Party in the *IAM User Guide*. You can retrieve the ARN from the Account A by opening the IAM console.

To find the ARN Value in Account A with the IAM console

1. Choose **Roles**
2. Choose the role you want to examine.
3. Look for the **Role ARN** in the **Summary** section.

The AssumeRole API returns temporary credentials that a user in accounts B, C, or Z can use to access resources in Account A. In this example, the resources you want to access are the Amazon S3 bucket and the log files that the bucket contains. The temporary credentials have the permissions that you defined in the role access policy.

The following Python example (using the Amazon SDK for Python (Boto)) shows how to call `AssumeRole` and how to use the temporary security credentials returned to list all Amazon S3 buckets controlled by Account A.

```
def list_buckets_from_assumed_role(user_key, assume_role_arn, session_name):
    """
    Assumes a role that grants permission to list the Amazon S3 buckets in the account.
    Uses the temporary credentials from the role to list the buckets that are owned
    by the assumed role's account.

    :param user_key: The access key of a user that has permission to assume the role.
    :param assume_role_arn: The Amazon Resource Name (ARN) of the role that
                            grants access to list the other account's buckets.
    :param session_name: The name of the STS session.
    """
    sts_client = boto3.client(
        'sts', aws_access_key_id=user_key.id, aws_secret_access_key=user_key.secret)
    try:
        response = sts_client.assume_role(
            RoleArn=assume_role_arn, RoleSessionName=session_name)
        temp_credentials = response['Credentials']
        print(f"Assumed role {assume_role_arn} and got temporary credentials.")
    except ClientError as error:
        print(f"Couldn't assume role {assume_role_arn}. Here's why: "
              f"{error.response['Error']['Message']}")
        raise

    # Create an S3 resource that can access the account with the temporary credentials.
    s3_resource = boto3.resource(
        's3',
        aws_access_key_id=temp_credentials['AccessKeyId'],
```

```
        aws_secret_access_key=temp_credentials['SecretAccessKey'],
        aws_session_token=temp_credentials['SessionToken'])
    print(f"Listing buckets for the assumed role's account:")
    try:
        for bucket in s3_resource.buckets.all():
            print(bucket.name)
    except ClientError as error:
        print(f"Couldn't list buckets for the account. Here's why: "
              f"{error.response['Error']['Message']}")
        raise
```

Stop sharing CloudTrail log files between Amazon accounts

To stop sharing log files to another Amazon Web Services account, you delete the role that you created for that account in Creating a role (p. 300).

1. Sign in to the Amazon Web Services Management Console with administrative permissions for Account A.
2. Navigate to the IAM console.
3. In the navigation pane, choose **Roles**.
4. Select the role you want to delete.
5. Choose **Delete**.

Validating CloudTrail log file integrity

To determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it, you can use CloudTrail log file integrity validation. This feature is built using industry standard algorithms: SHA-256 for hashing and SHA-256 with RSA for digital signing. This makes it computationally infeasible to modify, delete or forge CloudTrail log files without detection. You can use the Amazon CLI to validate the files in the location where CloudTrail delivered them.

Why use it?

Validated log files are invaluable in security and forensic investigations. For example, a validated log file enables you to assert positively that the log file itself has not changed, or that particular user credentials performed specific API activity. The CloudTrail log file integrity validation process also lets you know if a log file has been deleted or changed, or assert positively that no log files were delivered to your account during a given period of time.

How it works

When you enable log file integrity validation, CloudTrail creates a hash for every log file that it delivers. Every hour, CloudTrail also creates and delivers a file that references the log files for the last hour and contains a hash of each. This file is called a digest file. CloudTrail signs each digest file using the private key of a public and private key pair. After delivery, you can use the public key to validate the digest file. CloudTrail uses different key pairs for each Amazon Web Services Region.

The digest files are delivered to the same Amazon S3 bucket associated with your trail as your CloudTrail log files. If your log files are delivered from all Regions or from multiple accounts into a single Amazon S3 bucket, CloudTrail will deliver the digest files from those Regions and accounts into the same bucket.

The digest files are put into a folder separate from the log files. This separation of digest files and log files enables you to enforce granular security policies and permits existing log processing solutions to continue to operate without modification. Each digest file also contains the digital signature of the previous digest file if one exists. The signature for the current digest file is in the metadata properties of the digest file Amazon S3 object. For more information about digest file contents, see CloudTrail digest file structure (p. 314).

Storing log and digest files

You can store the CloudTrail log files and digest files in Amazon S3 or S3 Glacier securely, durably and inexpensively for an indefinite period of time. To enhance the security of the digest files stored in Amazon S3, you can use Amazon S3 MFA Delete.

Enabling validation and validating files

To enable log file integrity validation, you can use the Amazon Web Services Management Console, the Amazon CLI, or CloudTrail API. Enabling log file integrity validation allows CloudTrail to deliver digest log files to your Amazon S3 bucket, but does not validate the integrity of the files. For more information, see Enabling log file integrity validation for CloudTrail (p. 308).

To validate the integrity of CloudTrail log files, you can use the Amazon CLI or create your own solution. The Amazon CLI will validate files in the location where CloudTrail delivered them. If you want to validate logs that you have moved to a different location, either in Amazon S3 or elsewhere, you can create your own validation tools.

For information on validating logs by using the Amazon CLI, see Validating CloudTrail log file integrity with the Amazon CLI (p. 309). For information on developing custom implementations of CloudTrail log file validation, see Custom implementations of CloudTrail log file integrity validation (p. 319).

Enabling log file integrity validation for CloudTrail

You can enable log file integrity validation by using the Amazon Web Services Management Console, Amazon Command Line Interface (Amazon CLI), or CloudTrail API. CloudTrail starts delivering digest files in about an hour.

Amazon Web Services Management Console

To enable log file integrity validation with the CloudTrail console, choose **Yes** for the **Enable log file validation** option when you create or update a trail. By default, this feature is enabled for new trails. For more information, see Creating and updating a trail with the console (p. 78).

Amazon CLI

To enable log file integrity validation with the Amazon CLI, use the `--enable-log-file-validation` option with the create-trail or update-trail commands. To disable log file integrity validation, use the `--no-enable-log-file-validation` option.

Example

The following `update-trail` command enables log file validation and starts delivering digest files to the Amazon S3 bucket for the specified trail.

```
aws cloudtrail update-trail --name your-trail-name --enable-log-file-validation
```

CloudTrail API

To enable log file integrity validation with the CloudTrail API, set the `EnableLogFileValidation` request parameter to `true` when calling `CreateTrail` or `UpdateTrail`.

For more information, see CreateTrail and UpdateTrail in the Amazon CloudTrail API Reference.

Validating CloudTrail log file integrity with the Amazon CLI

To validate logs with the Amazon Command Line Interface, use the CloudTrail `validate-logs` command. The command uses the digest files delivered to your Amazon S3 bucket to perform the validation. For information about digest files, see CloudTrail digest file structure (p. 314).

The Amazon CLI allows you to detect the following types of changes:

- Modification or deletion of CloudTrail log files
- Modification or deletion of CloudTrail digest files
- Modification or deletion of both of the above

> **Note**
> The Amazon CLI validates only log files that are referenced by digest files. For more information, see Checking whether a particular file was delivered by CloudTrail (p. 314).

Prerequisites

To validate log file integrity with the Amazon CLI, the following conditions must be met:

- You must have online connectivity to Amazon.
- You must have read access to the Amazon S3 bucket that contains the digest and log files.
- The digest and log files must not have been moved from the original Amazon S3 location where CloudTrail delivered them.

> **Note**
> Log files that have been downloaded to local disk cannot be validated with the Amazon CLI. For guidance on creating your own tools for validation, see Custom implementations of CloudTrail log file integrity validation (p. 319).

validate-logs

Syntax

The following is the syntax for `validate-logs`. Optional parameters are shown in brackets.

```
aws cloudtrail validate-logs --trail-arn <trailARN> --start-time <start-time>
[--end-time <end-time>] [--s3-bucket <bucket-name>] [--s3-prefix <prefix>] [--
account-id <account-id>] [--verbose]
```

> **Note**
> The `validate-logs` command is Region specific. You must specify the `--region` global option to validate logs for a specific Amazon Web Services Region.

Options

The following are the command-line options for `validate-logs`. The `--trail-arn` and `--start-time` options are required. The `--account-id` option is additionally required for organizational trails.

`--start-time`

Specifies that log files delivered on or after the specified UTC timestamp value will be validated. Example: `2015-01-08T05:21:42Z`.

`--end-time`

Optionally specifies that log files delivered on or before the specified UTC timestamp value will be validated. The default value is the current UTC time (`Date.now()`). Example: `2015-01-08T12:31:41Z`.

Note
For the time range specified, the `validate-logs` command checks only the log files that are referenced in their corresponding digest files. No other log files in the Amazon S3 bucket are checked. For more information, see Checking whether a particular file was delivered by CloudTrail (p. 314).

`--s3-bucket`

Optionally specifies the Amazon S3 bucket where the digest files are stored. If a bucket name is not specified, the Amazon CLI will retrieve it by calling `DescribeTrails()`.

`--s3-prefix`

Optionally specifies the Amazon S3 prefix where the digest files are stored. If not specified, the Amazon CLI will retrieve it by calling `DescribeTrails()`.

Note
You should use this option only if your current prefix is different from the prefix that was in use during the time range that you specify.

`--account-id`

Optionally specifies the account for validating logs. This parameter is required for organization trails for validating logs for the specific account inside an organization.

`--trail-arn`

Specifies the Amazon Resource Name (ARN) of the trail to be validated. The format of a trail ARN follows.

```
arn:aws:cloudtrail:us-east-2:111111111111:trail/MyTrailName
```

Note
To obtain the trail ARN for a trail, you can use the `describe-trails` command before running `validate-logs`.
You may want to specify the bucket name and prefix in addition to the trail ARN if log files have been delivered to more than one bucket in the time range that you specified, and you want to restrict the validation to the log files in only one of the buckets.

`--verbose`

Optionally outputs validation information for every log or digest file in the specified time range. The output indicates whether the file remains unchanged or has been modified or deleted. In non-verbose mode (the default), information is returned only for those cases in which there was a validation failure.

Example

The following example validates log files from the specified start time to the present, using the Amazon S3 bucket configured for the current trail and specifying verbose output.

```
aws cloudtrail validate-logs --start-time 2015-08-27T00:00:00Z --end-time
 2015-08-28T00:00:00Z --trail-arn arn:aws:cloudtrail:us-east-2:111111111111:trail/my-trail-
name --verbose
```

How `validate-logs` works

The `validate-logs` command starts by validating the most recent digest file in the specified time range. First, it verifies that the digest file has been downloaded from the location to which it claims to belong. In other words, if the CLI downloads digest file `df1` from the S3 location `p1`, validate-logs will verify that `p1 == df1.digestS3Bucket + '/' + df1.digestS3Object`.

If the signature of the digest file is valid, it checks the hash value of each of the logs referenced in the digest file. The command then goes back in time, validating the previous digest files and their referenced log files in succession. It continues until the specified value for `start-time` is reached, or until the digest chain ends. If a digest file is missing or not valid, the time range that cannot be validated is indicated in the output.

Validation results

Validation results begin with a summary header in the following format:

```
Validating log files for trail trail_ARN  between time_stamp and time_stamp
```

Each line of the main output contains the validation results for a single digest or log file in the following format:

```
<Digest file | Log file> <S3 path> <Validation Message>
```

The following table describes the possible validation messages for log and digest files.

File Type	Validation Message	Description
`Digest file`	`valid`	The digest file signature is valid. The log files it references can be checked. This message is included only in verbose mode.
`Digest file`	`INVALID: has been moved from its original location`	The S3 bucket or S3 object from which the digest file was retrieved do not match the S3 bucket or S3 object locations that are recorded in the digest file itself.
`Digest file`	`INVALID: invalid format`	The format of the digest file is invalid. The log files corresponding to the time range that the digest file represents cannot be validated.

File Type	Validation Message	Description
`Digest file`	`INVALID: not found`	The digest file was not found. The log files corresponding to the time range that the digest file represents cannot be validated.
`Digest file`	`INVALID: public key not found for fingerprint` *`fingerprint`*	The public key corresponding to the fingerprint recorded in the digest file was not found. The digest file cannot be validated.
`Digest file`	`INVALID: signature verification failed`	The digest file signature is not valid. Because the digest file is not valid, the log files it references cannot be validated, and no assertions can be made about the API activity in them.
`Digest file`	`INVALID: Unable to load PKCS #1 key with fingerprint` *`fingerprint`*	Because the DER encoded public key in PKCS #1 format having the specified fingerprint could not be loaded, the digest file cannot be validated.
`Log file`	`valid`	The log file has been validated and has not been modified since the time of delivery. This message is included only in verbose mode.
`Log file`	`INVALID: hash value doesn't match`	The hash for the log file does not match. The log file has been modified after delivery by CloudTrail.
`Log file`	`INVALID: invalid format`	The format of the log file is invalid. The log file cannot be validated.
`Log file`	`INVALID: not found`	The log file was not found and cannot be validated.

The output includes summary information about the results returned.

Example outputs

Verbose

The following example `validate-logs` command uses the `--verbose` flag and produces the sample output that follows. `[...]` indicates the sample output has been abbreviated.

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-2:111111111111:trail/
example-trail-name --start-time 2015-08-31T22:00:00Z --end-time 2015-09-01T19:17:29Z --
verbose
```

```
Validating log files for trail arn:aws:cloudtrail:us-east-2:111111111111:trail/example-
trail-name between 2015-08-31T22:00:00Z and 2015-09-01T19:17:29Z

Digest file    s3://example-bucket/AWSLogs/111111111111/CloudTrail-Digest/us-
east-2/2015/09/01/111111111111_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T201728Z.json.gz valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1925Z_WZZw1RymnjCRjxXc.json.gz
 valid
```

```
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1915Z_POuvV87nu6pfAV2W.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1930Z_l2QgXhAKVm1QXiIA.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1920Z_eQJteBBrfpBCqOqw.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1950Z_9g5A6qlR2B5KaRdq.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1920Z_i4DNCC12BuXd6Ru7.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1915Z_Sg5caf2RH6Jdx0EJ.json.gz
 valid
Digest file    s3://example-bucket/AWSLogs/111111111111/CloudTrail-Digest/us-
east-2/2015/09/01/111111111111_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T191728Z.json.gz valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/09/01/111111111111_CloudTrail_us-east-2_20150901T1910Z_YYSFiuFQk4nrtnEW.json.gz
 valid
[...]
Log file       s3://example-bucket/AWSLogs/144218288521/CloudTrail/us-
east-2/2015/09/01/144218288521_CloudTrail_us-east-2_20150901T1055Z_0Sfy6m9f6iBzmoPF.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/144218288521/CloudTrail/us-
east-2/2015/09/01/144218288521_CloudTrail_us-east-2_20150901T1040Z_lLa3QzVLpOed7igR.json.gz
 valid

Digest file    s3://example-bucket/AWSLogs/144218288521/CloudTrail-Digest/us-
east-2/2015/09/01/144218288521_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T101728Z.json.gz INVALID: signature verification failed

Digest file    s3://example-bucket/AWSLogs/144218288521/CloudTrail-Digest/us-
east-2/2015/09/01/144218288521_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T091728Z.json.gz valid
Log file       s3://example-bucket/AWSLogs/144218288521/CloudTrail/us-
east-2/2015/09/01/144218288521_CloudTrail_us-east-2_20150901T0830Z_eaFvO3dwHo4NCqqc.json.gz
 valid
Digest file    s3://example-bucket/AWSLogs/144218288521/CloudTrail-Digest/us-
east-2/2015/09/01/144218288521_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T081728Z.json.gz valid
Digest file    s3://example-bucket/AWSLogs/144218288521/CloudTrail-Digest/us-
east-2/2015/09/01/144218288521_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T071728Z.json.gz valid
[...]
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/08/31/111111111111_CloudTrail_us-east-2_20150831T2245Z_mbJkEO5kNcDnVhGh.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/08/31/111111111111_CloudTrail_us-east-2_20150831T2225Z_IQ6kXy8sKU03RSPr.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/08/31/111111111111_CloudTrail_us-east-2_20150831T2230Z_eRPVRTxHQ5498ROA.json.gz
 valid
Log file       s3://example-bucket/AWSLogs/111111111111/CloudTrail/us-
east-2/2015/08/31/111111111111_CloudTrail_us-east-2_20150831T2255Z_IlWawYZGvTWB5vYN.json.gz
 valid
Digest file    s3://example-bucket/AWSLogs/111111111111/CloudTrail-Digest/us-
east-2/2015/08/31/111111111111_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150831T221728Z.json.gz valid

Results requested for 2015-08-31T22:00:00Z to 2015-09-01T19:17:29Z
```

```
Results found for 2015-08-31T22:17:28Z to 2015-09-01T20:17:28Z:

22/23 digest files valid, 1/23 digest files INVALID
63/63 log files valid
```

Non-verbose

The following example `validate-logs` command does not use the `--verbose` flag. In the sample output that follows, one error was found. Only the header, error, and summary information are returned.

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-2:111111111111:trail/
example-trail-name --start-time 2015-08-31T22:00:00Z --end-time 2015-09-01T19:17:29Z
```

```
Validating log files for trail arn:aws:cloudtrail:us-east-2:111111111111:trail/example-
trail-name between 2015-08-31T22:00:00Z and 2015-09-01T19:17:29Z

Digest file s3://example-bucket/AWSLogs/144218288521/CloudTrail-Digest/us-
east-2/2015/09/01/144218288521_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150901T101728Z.json.gz INVALID: signature verification failed

Results requested for 2015-08-31T22:00:00Z to 2015-09-01T19:17:29Z
Results found for 2015-08-31T22:17:28Z to 2015-09-01T20:17:28Z:

22/23 digest files valid, 1/23 digest files INVALID
63/63 log files valid
```

Checking whether a particular file was delivered by CloudTrail

To check if a particular file in your bucket was delivered by CloudTrail, run `validate-logs` in verbose mode for the time period that includes the file. If the file appears in the output of `validate-logs`, then the file was delivered by CloudTrail.

CloudTrail digest file structure

Each digest file contains the names of the log files that were delivered to your Amazon S3 bucket during the last hour, the hash values for those log files, and the digital signature of the previous digest file. The signature for the current digest file is stored in the metadata properties of the digest file object. The digital signatures and hashes are used for validating the integrity of the log files and of the digest file itself.

Digest file location

Digest files are delivered to an Amazon S3 bucket location that follows this syntax.

```
s3://s3-bucket-name/optional-prefix/AWSLogs/aws-account-id/CloudTrail-Digest/
    region/digest-end-year/digest-end-month/digest-end-date/
    aws-account-id_CloudTrail-Digest_region_trail-name_region_digest_end_timestamp.json.gz
```

Note
For organization trails, the bucket location also includes the organization unit ID, as follows:

```
s3://s3-bucket-name/optional-prefix/AWSLogs/O-ID/aws-account-id/CloudTrail-Digest/
    region/digest-end-year/digest-end-month/digest-end-date/
    aws-account-id_CloudTrail-Digest_region_trail-
name_region_digest_end_timestamp.json.gz
```

Sample digest file contents

The following example digest file contains information for a CloudTrail log.


```
{
  "awsAccountId": "111122223333",
  "digestStartTime": "2015-08-17T14:01:31Z",
  "digestEndTime": "2015-08-17T15:01:31Z",
  "digestS3Bucket": "S3-bucket-name",
  "digestS3Object": "AWSLogs/111122223333/CloudTrail-Digest/us-
east-2/2015/08/17/111122223333_CloudTrail-Digest_us-east-2_your-trail-name_us-
east-2_20150817T150131Z.json.gz",
  "digestPublicKeyFingerprint": "31e8b5433410dfb61a9dc45cc65b22ff",
  "digestSignatureAlgorithm": "SHA256withRSA",
  "newestEventTime": "2015-08-17T14:52:27Z",
  "oldestEventTime": "2015-08-17T14:42:27Z",
  "previousDigestS3Bucket": "S3-bucket-name",
  "previousDigestS3Object": "AWSLogs/111122223333/CloudTrail-Digest/us-
east-2/2015/08/17/111122223333_CloudTrail-Digest_us-east-2_your-trail-name_us-
east-2_20150817T140131Z.json.gz",
  "previousDigestHashValue":
 "97fb791cf91ffc440d274f8190dbdd9aa09c34432aba82739df18b6d3c13df2d",
  "previousDigestHashAlgorithm": "SHA-256",
  "previousDigestSignature":
 "50887ccffad4c002b97caa37cc9dc626e3c680207d41d27fa5835458e066e0d3652fc4dfc30937e4d5f4cc7f796e7a258fb50
  "logFiles": [
    {
      "s3Bucket": "S3-bucket-name",
      "s3Object": "AWSLogs/111122223333/CloudTrail/us-
east-2/2015/08/17/111122223333_CloudTrail_us-
east-2_20150817T1445Z_9nYN7gp2eWAJHIfT.json.gz",
      "hashValue": "9bb6196fc6b84d6f075a56548feca262bd99ba3c2de41b618e5b6e22c1fc71f6",
      "hashAlgorithm": "SHA-256",
      "newestEventTime": "2015-08-17T14:52:27Z",
      "oldestEventTime": "2015-08-17T14:42:27Z"
    }
  ]
}
```


Digest file field descriptions

The following are descriptions for each field in the digest file:

`awsAccountId`

The Amazon account ID for which the digest file has been delivered.

`digestStartTime`

The starting UTC time range that the digest file covers, taking as a reference the time in which log files have been delivered by CloudTrail. This means that if the time range is [Ta, Tb], the digest will contain all the log files delivered to the customer between Ta and Tb.

`digestEndTime`

The ending UTC time range that the digest file covers, taking as a reference the time in which log files have been delivered by CloudTrail. This means that if the time range is [Ta, Tb], the digest will contain all the log files delivered to the customer between Ta and Tb.

`digestS3Bucket`

The name of the Amazon S3 bucket to which the current digest file has been delivered.

`digestS3Object`

The Amazon S3 object key (that is, the Amazon S3 bucket location) of the current digest file. The first two Regions in the string show the Region from which the digest file was delivered. The last Region (after `your-trail-name`) is the home Region of the trail. The home Region is the Region in which the trail was created. In the case of a multi-Region trail, this can be different from the Region from which the digest file was delivered.

`newestEventTime`

The UTC time of the most recent event among all of the events in the log files in the digest.

`oldestEventTime`

The UTC time of the oldest event among all of the events in the log files in the digest.

Note
If the digest file is delivered late, the value of `oldestEventTime` will be earlier than the value of `digestStartTime`.

`previousDigestS3Bucket`

The Amazon S3 bucket to which the previous digest file was delivered.

`previousDigestS3Object`

The Amazon S3 object key (that is, the Amazon S3 bucket location) of the previous digest file.

`previousDigestHashValue`

The hexadecimal encoded hash value of the uncompressed contents of the previous digest file.

`previousDigestHashAlgorithm`

The name of the hash algorithm that was used to hash the previous digest file.

`publicKeyFingerprint`

The hexadecimal encoded fingerprint of the public key that matches the private key used to sign this digest file. You can retrieve the public keys for the time range corresponding to the digest file by using the Amazon CLI or the CloudTrail API. Of the public keys returned, the one whose fingerprint matches this value can be used for validating the digest file. For information about retrieving public keys for digest files, see the Amazon CLI `list-public-keys` command or the CloudTrail `ListPublicKeys` API.

Note
CloudTrail uses different private/public key pairs per Region. Each digest file is signed with a private key unique to its Region. Therefore, when you validate a digest file from a particular Region, you must look in the same Region for its corresponding public key.

`digestSignatureAlgorithm`

The algorithm used to sign the digest file.

`logFiles.s3Bucket`

The name of the Amazon S3 bucket for the log file.

`logFiles.s3Object`

The Amazon S3 object key of the current log file.

`logFiles.newestEventTime`

The UTC time of the most recent event in the log file. This time also corresponds to the time stamp of the log file itself.

`logFiles.oldestEventTime`

The UTC time of the oldest event in the log file.

`logFiles.hashValue`

The hexadecimal encoded hash value of the uncompressed log file content.

`logFiles.hashAlgorithm`

The hash algorithm used to hash the log file.

Starting digest file

When log file integrity validation is started, a starting digest file will be generated. A starting digest file will also be generated when log file integrity validation is restarted (by either disabling and then reenabling log file integrity validation, or by stopping logging and then restarting logging with validation enabled). In a starting digest file, the following fields relating to the previous digest file will be null:

- `previousDigestS3Bucket`
- `previousDigestS3Object`
- `previousDigestHashValue`
- `previousDigestHashAlgorithm`
- `previousDigestSignature`

'Empty' digest files

CloudTrail will deliver a digest file even when there has been no API activity in your account during the one hour period that the digest file represents. This can be useful when you need to assert that no log files were delivered during the hour reported by the digest file.

The following example shows the contents of a digest file that recorded an hour when no API activity occurred. Note that the `logFiles:[ ]` field at the end of the digest file contents is empty.

```
{
  "awsAccountId": "111122223333",
  "digestStartTime": "2015-08-20T17:01:31Z",
  "digestEndTime": "2015-08-20T18:01:31Z",
  "digestS3Bucket": "example-bucket-name",
  "digestS3Object": "AWSLogs/111122223333/CloudTrail-Digest/us-
east-2/2015/08/20/111122223333_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150820T180131Z.json.gz",
  "digestPublicKeyFingerprint": "31e8b5433410dfb61a9dc45cc65b22ff",
  "digestSignatureAlgorithm": "SHA256withRSA",
  "newestEventTime": null,
  "oldestEventTime": null,
  "previousDigestS3Bucket": "example-bucket-name",
  "previousDigestS3Object": "AWSLogs/111122223333/CloudTrail-Digest/us-
east-2/2015/08/20/111122223333_CloudTrail-Digest_us-east-2_example-trail-name_us-
east-2_20150820T170131Z.json.gz",
  "previousDigestHashValue":
 "ed96c4bac9eaa8fe9716ca0e515da51938be651b1db31d781956416a9d05cdfa",
  "previousDigestHashAlgorithm": "SHA-256",
  "previousDigestSignature":
 "82705525fb0fe7f919f9434e5b7138cb41793c776c7414f3520c0242902daa8cc8286b29263d2627f2f259471c745b1654af7
  "logFiles": []
}
```

Signature of the digest file

The signature information for a digest file is located in two object metadata properties of the Amazon S3 digest file object. Each digest file has the following metadata entries:

- `x-amz-meta-signature`

  The hexadecimal encoded value of the digest file signature. The following is an example signature:

```
3be472336fa2989ef34de1b3c1bf851f59eb030eaff3e2fb6600a082a23f4c6a82966565b994f9de4a5989d053d9d15d20fc5
28f1cc237f372264a51b611c01da429565def703539f4e71009051769469231bc22232fa260df02740047af532229885ea2b0
05d3ffcb5d2dd5dc28f8bb5b7993938e8a5f912a82b448a367eccb2ec0f198ba71e23eb0b97278cf65f3c8d1e652c6de33a22
```

- `x-amz-meta-signature-algorithm`

  The following shows an example value of the algorithm used to generate the digest signature:

  `SHA256withRSA`

Digest file chaining

The fact that each digest file contains a reference to its previous digest file enables a "chaining" that permits validation tools like the Amazon CLI to detect if a digest file has been deleted. It also allows the digest files in a specified time range to be successively inspected, starting with the most recent first.

> **Note**
> When you disable log file integrity validation, the chain of digest files is broken after one hour. CloudTrail will not create digest files for log files that were delivered during a period in which log file integrity validation was disabled. For example, if you enable log file integrity validation at noon on January 1, disable it at noon on January 2, and re-enable it at noon on January 10, digest files will not be created for the log files delivered from noon on January 2 to noon on January 10. The same applies whenever you stop CloudTrail logging or delete a trail.

If logging is stopped or the trail is deleted, CloudTrail will deliver a final digest file. This digest file can contain information for any remaining log files that cover events up to and including the `StopLogging` event.

Custom implementations of CloudTrail log file integrity validation

Because CloudTrail uses industry standard, openly available cryptographic algorithms and hash functions, you can create your own tools to validate the integrity of CloudTrail log files. When log file integrity validation is enabled, CloudTrail delivers digest files to your Amazon S3 bucket. You can use these files to implement your own validation solution. For more information about digest files, see CloudTrail digest file structure (p. 314).

This topic describes how digest files are signed, and then details the steps that you will need to take to implement a solution that validates the digest files and the log files that they reference.

Understanding how CloudTrail digest files are signed

CloudTrail digest files are signed with RSA digital signatures. For each digest file, CloudTrail does the following:

1. Creates a string for data signing based on designated digest file fields (described in the next section).
2. Gets a private key unique to the Region.
3. Passes the SHA-256 hash of the string and the private key to the RSA signing algorithm, which produces a digital signature.
4. Encodes the byte code of the signature into hexadecimal format.
5. Puts the digital signature into the `x-amz-meta-signature` metadata property of the Amazon S3 digest file object.

Contents of the data signing string

The following CloudTrail objects are included in the string for data signing:

- The ending timestamp of the digest file in UTC extended format (for example, `2015-05-08T07:19:37Z`)
- The current digest file S3 path
- The hexadecimal-encoded SHA-256 hash of the current digest file
- The hexadecimal-encoded signature of the previous digest file

The format for calculating this string and an example string are provided later in this document.

Custom validation implementation steps

When implementing a custom validation solution, you will need to validate the digest file first, and then the log files that it references.

Validate the digest file

To validate a digest file, you need its signature, the public key whose private key was used to signed it, and a data signing string that you compute.

1. Get the digest file.

2. Verify that the digest file has been retrieved from its original location.
3. Get the hexadecimal-encoded signature of the digest file.
4. Get the hexadecimal-encoded fingerprint of the public key whose private key was used to sign the digest file.
5. Retrieve the public keys for the time range corresponding to the digest file.
6. From among the public keys retrieved, choose the public key whose fingerprint matches the fingerprint in the digest file.
7. Using the digest file hash and other digest file fields, recreate the data signing string used to verify the digest file signature.
8. Validate the signature by passing in the SHA-256 hash of the string, the public key, and the signature as parameters to the RSA signature verification algorithm. If the result is true, the digest file is valid.

Validate the log files

If the digest file is valid, validate each of the log files that the digest file references.

1. To validate the integrity of a log file, compute its SHA-256 hash value on its uncompressed content and compare the results with the hash for the log file recorded in hexadecimal in the digest. If the hashes match, the log file is valid.
2. By using the information about the previous digest file that is included in the current digest file, validate the previous digest files and their corresponding log files in succession.

The following sections describe these steps in detail.

A. Get the digest file

The first steps are to get the most recent digest file, verify that you have retrieved it from its original location, verify its digital signature, and get the fingerprint of the public key.

1. Using `S3 Get` or the AmazonS3Client class (for example), get the most recent digest file from your Amazon S3 bucket for the time range that you want to validate.
2. Check that the S3 bucket and S3 object used to retrieve the file match the S3 bucket S3 object locations that are recorded in the digest file itself.
3. Next, get the digital signature of the digest file from the `x-amz-meta-signature` metadata property of the digest file object in Amazon S3.
4. In the digest file, get the fingerprint of the public key whose private key was used to sign the digest file from the `digestPublicKeyFingerprint` field.

B. Retrieve the public key for validating the digest file

To get the public key to validate the digest file, you can use either the Amazon CLI or the CloudTrail API. In both cases, you specify a time range (that is, a start time and end time) for the digest files that you want to validate. One or more public keys may be returned for the time range that you specify. The returned keys may have validity time ranges that overlap.

> **Note**
> Because CloudTrail uses different private/public key pairs per Region, each digest file is signed with a private key unique to its Region. Therefore, when you validate a digest file from a particular Region, you must retrieve its public key from the same Region.

Use the Amazon CLI to retrieve public keys

To retrieve public keys for digest files by using the Amazon CLI, use the `cloudtrail list-public-keys` command. The command has the following format:

```
aws cloudtrail list-public-keys [--start-time <start-time>] [--end-time <end-
time>]
```

The start-time and end-time parameters are UTC timestamps and are optional. If not specified, the current time is used, and the currently active public key or keys are returned.

Sample Response

The response will be a list of JSON objects representing the key (or keys) returned:

```
{
    "publicKeyList": [
        {
            "ValidityStartTime": "1436317441.0",
            "ValidityEndTime": "1438909441.0",
            "Value": "MIIBCgKCAQEAn11L2YZ9h7onug2ILi1MWyHiMRsTQjfWE
+pHVRLk1QjfWhirG+lpOa8NrwQ/r7Ah5bNL6HepznOU9XTDSfmmnP97mqyc7z/upfZdS/AHhYcGaz7n6Wc/
RRBU6VmiPCrAUojuSk6/GjvA8iOPFsYDuBtviXarvuLPlrT9kAd4Lb+rFfR5peEgBEkhlzc5HuWO7S0y
+KunqxX6jQBnXGMtxmPBPP0FylgWGNdFtks/4YSKcgqwH0YDcawP9GGGDAeCIqPWIXDLG1jOjRRzWfCmD0iJUkz8vTsn4hq/5ZxRFE7
            "Fingerprint": "8eba5db5bea9b640d1c96a77256fe7f2"
        },
        {
            "ValidityStartTime": "1434589460.0",
            "ValidityEndTime": "1437181460.0",
            "Value": "MIIBCgKCAQEApfYL2FiZhpN74LNWVUzhR
+VheYhwhYm8w0n5Gf6i95ylW5kBAWKVEmnAQG7BvS5g9SMqFDQx52fW7NWV44IvfJ2xGXT
+wT+DgR6ZQ+6yxskQNqV5YcXj4Aa5Zz4jJfsYjDuO2MDTZNIzNvBNzaBJ+r2WIWAJ/
Xq54kyF63B6WE38vKuDE7nSd1FqQuEoNBFLPInvgggYe2Ym1Refe2z71wNcJ2kY
+q0h1BSHrSM8RWuJIw7MXwF9iQncg9jYzUlNJomozQzAG5wSRfbplcCYNY40xvGd/aAmO0m+Y
+XFMrKwtLCwseHPvj843qVno6x4BJN9bpWnoPo9sdsbGoiK3QIDAQAB",
            "Fingerprint": "8933b39ddc64d26d8e14ffbf6566fee4"
        },
        {
            "ValidityStartTime": "1434589370.0",
            "ValidityEndTime": "1437181370.0",
            "Value":
 "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqlzPJbvZJ42UdcmLfPUqXYNfOs6I8lCfao/
tOs8CmzPOEdtLWugB9xoIUz78qVHdKIqxbaG4jWHfJBiOSSFBM0lt8cdVo4TnRa7oG9io5pysS6DJhBBAeXsicufsiFJR
+wrUNh8RSLxL4k6G1+BhLX20tJkZ/erT97tDGBujAelqseGg3vPZbTx9SMfOLN65PdLFudLP7Gat0Z9p5jw/
rjpclKfo9Bfc3heeBxWGKwBBOKnFAaN9V57pOaosCvPKmHd9bg7jsQkI9Xp22IzGLsTFJZYVA3KiTAElDMu80iFXPHEq9hKNbt9e4UP
+1utKVEiLkR2disdCmPTK0VQIDAQAB",
            "Fingerprint": "31e8b5433410dfb61a9dc45cc65b22ff"
        }
    ]
}
```

Use the CloudTrail API to retrieve public keys

To retrieve public keys for digest files by using the CloudTrail API, pass in start time and end time values to the `ListPublicKeys` API. The `ListPublicKeys` API returns the public keys whose private keys were used to sign digest files within the specified time range. For each public key, the API also returns the corresponding fingerprint.

`ListPublicKeys`

This section describes the request parameters and response elements for the `ListPublicKeys` API.

> **Note**
> The encoding for the binary fields for `ListPublicKeys` is subject to change.

Request Parameters

Name	Description
`StartTime`	Optionally specifies, in UTC, the start of the time range to look up public keys for CloudTrail digest files. If StartTime is not specified, the current time is used, and the current public key is returned. Type: DateTime
`EndTime`	Optionally specifies, in UTC, the end of the time range to look up public keys for CloudTrail digest files. If EndTime is not specified, the current time is used. Type: DateTime

Response Elements

`PublicKeyList`, an array of `PublicKey` objects that contains:

Name	Description
`Value`	The DER encoded public key value in PKCS #1 format. Type: Blob
`ValidityStartTime`	The starting time of validity of the public key. Type: DateTime
`ValidityEndTime`	The ending time of validity of the public key. Type: DateTime
`Fingerprint`	The fingerprint of the public key. The fingerprint can be used to identify the public key that you must use to validate the digest file. Type: String

C. Choose the public key to use for validation

From among the public keys retrieved by `list-public-keys` or `ListPublicKeys`, choose the public key returned whose fingerprint matches the fingerprint recorded in the `digestPublicKeyFingerprint` field of the digest file. This is the public key that you will use to validate the digest file.

D. Recreate the data signing string

Now that you have the signature of the digest file and associated public key, you need to calculate the data signing string. After you have calculated the data signing string, you will have the inputs needed to verify the signature.

The data signing string has the following format:

```
Data_To_Sign_String =
  Digest_End_Timestamp_in_UTC_Extended_format + '\n' +
  Current_Digest_File_S3_Path + '\n' +
  Hex(Sha256(current-digest-file-content)) + '\n' +
  Previous_digest_signature_in_hex
```

An example `Data_To_Sign_String` follows.

```
2015-08-12T04:01:31Z
S3-bucket-name/AWSLogs/111122223333/CloudTrail-Digest/us-east-2/2015/08/12/111122223333_us-
east-2_CloudTrail-Digest_us-east-2_20150812T040131Z.json.gz
4ff08d7c6ecd6eb313257e839645d20363ee3784a2328a7d76b99b53cc9bcacd
6e8540b83c3ac86a0312d971a225361d28ed0af20d70c211a2d405e32abf529a8145c2966e3bb47362383a52441545ed091fb81
d4c7c09dd152b84e79099ce7a9ec35d2b264eb92eb6e090f1e5ec5d40ec8a0729c02ff57f9e30d5343a8591638f8b794972ce15
98b0aee2c1c8af74ec620261529265e83a9834ebef6054979d3e9a6767dfa6fdb4ae153436c567d6ae208f988047ccfc8e5e41f
```

After you recreate this string, you can validate the digest file.

E. Validate the digest file

Pass the SHA-256 hash of the recreated data signing string, digital signature, and public key to the RSA signature verification algorithm. If the output is true, the signature of the digest file is verified and the digest file is valid.

F. Validate the log files

After you have validated the digest file, you can validate the log files it references. The digest file contains the SHA-256 hashes of the log files. If one of the log files was modified after CloudTrail delivered it, the SHA-256 hashes will change, and the signature of digest file will not match.

The following shows how validate the log files:

1. Do an `S3 Get` of the log file using the S3 location information in the digest file's `logFiles.s3Bucket` and `logFiles.s3Object` fields.
2. If the `S3 Get` operation is successful, iterate through the log files listed in the digest file's logFiles array using the following steps:
   a. Retrieve the original hash of the file from the `logFiles.hashValue` field of the corresponding log in the digest file.
   b. Hash the uncompressed contents of the log file with the hashing algorithm specified in `logFiles.hashAlgorithm`.
   c. Compare the hash value that you generated with the one for the log in the digest file. If the hashes match, the log file is valid.

G. Validate additional digest and log files

In each digest file, the following fields provide the location and signature of the previous digest file:

- `previousDigestS3Bucket`
- `previousDigestS3Object`
- `previousDigestSignature`

Use this information to visit previous digest files sequentially, validating the signature of each and the log files that they reference by using the steps in the previous sections. The only difference is that for previous digest files, you do not need to retrieve the digital signature from the digest file object's Amazon S3 metadata properties. The signature for the previous digest file is provided for you in the `previousDigestSignature` field.

You can go back until the starting digest file is reached, or until the chain of digest files is broken, whichever comes first.

Validating digest and log files offline

When validating digest and log files offline, you can generally follow the procedures described in the previous sections. However, you must take into account the following areas:

Handling the most recent digest file

The digital signature of the most recent (that is, "current") digest file is in the Amazon S3 metadata properties of the digest file object. In an offline scenario, the digital signature for the current digest file will not be available.

Two possible ways of handling this are:

- Since the digital signature for the previous digest file is in the current digest file, start validating from the next-to-last digest file. With this method, the most recent digest file cannot be validated.
- As a preliminary step, obtain the signature for the current digest file from the digest file object's metadata properties and then store it securely offline. This would allow the current digest file to be validated in addition to the previous files in the chain.

Path resolution

Fields in the downloaded digest files like `s3object` and `previousDigestS3object` will still be pointing to Amazon S3 online locations for log files and digest files. An offline solution must find a way to reroute these to the current path of the downloaded log and digest files.

Public keys

In order to validate offline, all of the public keys that you need for validating log files in a given time range must first be obtained online (by calling `ListPublicKeys`, for example) and then stored securely offline. This step must be repeated whenever you want to validate additional files outside the initial time range that you specified.

Sample validation snippet

The following sample snippet provides skeleton code for validating CloudTrail digest and log files. The skeleton code is online/offline agnostic; that is, it is up to you to decide whether to implement it with or without online connectivity to Amazon. The suggested implementation uses the Java Cryptography Extension (JCE) and Bouncy Castle as a security provider.

The sample snippet shows:

- How to create the data signing string used to validate the digest file signature.
- How to verify the digest file signature.
- How to verify the log file hashes.
- A code structure for validating a chain of digest files.

```
import java.util.Arrays;
import java.security.MessageDigest;
import java.security.KeyFactory;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
import java.security.spec.X509EncodedKeySpec;
import org.json.JSONObject;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.apache.commons.codec.binary.Hex;
```

```
public class DigestFileValidator {

    public void validateDigestFile(String digestS3Bucket, String digestS3Object, String
 digestSignature) {

        // Using the Bouncy Castle provider as a JCE security provider - http://
www.bouncycastle.org/
        Security.addProvider(new BouncyCastleProvider());

        // Load the digest file from S3 (using Amazon S3 Client) or from your local copy
        JSONObject digestFile = loadDigestFileInMemory(digestS3Bucket, digestS3Object);

        // Check that the digest file has been retrieved from its original location
        if (!digestFile.getString("digestS3Bucket").equals(digestS3Bucket) ||
                !digestFile.getString("digestS3Object").equals(digestS3Object)) {
            System.err.println("Digest file has been moved from its original location.");
        } else {
            // Compute digest file hash
            MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
            messageDigest.update(convertToByteArray(digestFile));
            byte[] digestFileHash = messageDigest.digest();
            messageDigest.reset();

            // Compute the data to sign
            String dataToSign = String.format("%s%n%s/%s%n%s%n%s",
                                digestFile.getString("digestEndTime"),
                                digestFile.getString("digestS3Bucket"),
 digestFile.getString("digestS3Object"), // Constructing the S3 path of the digest file as
 part of the data to sign
                                Hex.encodeHexString(digestFileHash),
                                digestFile.getString("previousDigestSignature"));

            byte[] signatureContent = Hex.decodeHex(digestSignature);

            /*
                NOTE:
                To find the right public key to verify the signature, call CloudTrail
 ListPublicKey API to get a list
                of public keys, then match by the publicKeyFingerprint in the digest file.
 Also, the public key bytes
                returned from ListPublicKey API are DER encoded in PKCS#1 format:

                PublicKeyInfo ::= SEQUENCE {
                    algorithm       AlgorithmIdentifier,
                    PublicKey       BIT STRING
                }

                AlgorithmIdentifier ::= SEQUENCE {
                    algorithm       OBJECT IDENTIFIER,
                    parameters      ANY DEFINED BY algorithm OPTIONAL
                }
            */
            pkcs1PublicKeyBytes =
 getPublicKey(digestFile.getString("digestPublicKeyFingerprint")));

            // Transform the PKCS#1 formatted public key to x.509 format.
            RSAPublicKey rsaPublicKey = RSAPublicKey.getInstance(pkcs1PublicKeyBytes);
            AlgorithmIdentifier rsaEncryption = new
 AlgorithmIdentifier(PKCSObjectIdentifiers.rsaEncryption, null);
            SubjectPublicKeyInfo publicKeyInfo = new SubjectPublicKeyInfo(rsaEncryption,
 rsaPublicKey);

            // Create the PublicKey object needed for the signature validation
            PublicKey publicKey = KeyFactory.getInstance("RSA", "BC").generatePublic(new
 X509EncodedKeySpec(publicKeyInfo.getEncoded()));
```

```
            // Verify signature
            Signature signature = Signature.getInstance("SHA256withRSA", "BC");
            signature.initVerify(publicKey);
            signature.update(dataToSign.getBytes("UTF-8"));

            if (signature.verify(signatureContent)) {
                System.out.println("Digest file signature is valid, validating log
 files…");
                for (int i = 0; i < digestFile.getJSONArray("logFiles").length(); i++) {

                    JSONObject logFileMetadata =
 digestFile.getJSONArray("logFiles").getJSONObject(i);

                    // Compute log file hash
                    byte[] logFileContent = loadUncompressedLogFileInMemory(
                                                logFileMetadata.getString("s3Bucket"),
                                                logFileMetadata.getString("s3Object")
                                            );
                    messageDigest.update(logFileContent);
                     byte[] logFileHash = messageDigest.digest();
                    messageDigest.reset();

                    // Retrieve expected hash for the log file being processed
                    byte[] expectedHash =
 Hex.decodeHex(logFileMetadata.getString("hashValue"));

                    boolean signaturesMatch = Arrays.equals(expectedHash, logFileHash);
                    if (!signaturesMatch) {
                        System.err.println(String.format("Log file: %s/%s hash doesn't
 match.\tExpected: %s Actual: %s",
                                logFileMetadata.getString("s3Bucket"),
 logFileMetadata.getString("s3Object"),
                                Hex.encodeHexString(expectedHash),
 Hex.encodeHexString(logFileHash)));
                    } else {
                        System.out.println(String.format("Log file: %s/%s hash match",
                                logFileMetadata.getString("s3Bucket"),
 logFileMetadata.getString("s3Object")));
                    }
                }

            } else {
                System.err.println("Digest signature failed validation.");
            }

            System.out.println("Digest file validation completed.");

            if (chainValidationIsEnabled()) {
                // This enables the digests' chain validation
                validateDigestFile(
                        digestFile.getString("previousDigestS3Bucket"),
                        digestFile.getString("previousDigestS3Object"),
                        digestFile.getString("previousDigestSignature"));
            }
        }
    }
}
```

Using the CloudTrail Processing Library

The CloudTrail Processing Library is a Java library that provides an easy way to process Amazon CloudTrail logs. You provide configuration details about your CloudTrail SQS queue and write code to process events. The CloudTrail Processing Library does the rest. It polls your Amazon SQS queue, reads and parses queue messages, downloads CloudTrail log files, parses events in the log files, and passes the events to your code as Java objects.

The CloudTrail Processing Library is highly scalable and fault-tolerant. It handles parallel processing of log files so that you can process as many logs as needed. It handles network failures related to network timeouts and inaccessible resources.

The following topic shows you how to use the CloudTrail Processing Library to process CloudTrail logs in your Java projects.

The library is provided as an Apache-licensed open-source project, available on GitHub: https://github.com/aws/aws-cloudtrail-processing-library. The library source includes sample code that you can use as a base for your own projects.

Topics

- Minimum requirements (p. 327)
- Processing CloudTrail logs (p. 327)
- Advanced topics (p. 331)
- Additional resources (p. 335)

Minimum requirements

To use the CloudTrail Processing Library, you must have the following:

- Amazon SDK for Java 1.11.830
- Java 1.8 (Java SE 8)

Processing CloudTrail logs

To process CloudTrail logs in your Java application:

1. Adding the CloudTrail Processing Library to your project (p. 327)
2. Configuring the CloudTrail Processing Library (p. 329)
3. Implementing the events processor (p. 330)
4. Instantiating and running the processing executor (p. 331)

Adding the CloudTrail Processing Library to your project

To use the CloudTrail Processing Library, add it to your Java project's classpath.

Contents

- Adding the library to an Apache Ant project (p. 328)
- Adding the library to an Apache Maven project (p. 328)
- Adding the library to an Eclipse project (p. 328)
- Adding the library to an IntelliJ project (p. 329)

Adding the library to an Apache Ant project

To add the CloudTrail Processing Library to an Apache Ant project

1. Download or clone the CloudTrail Processing Library source code from GitHub:
   - https://github.com/aws/aws-cloudtrail-processing-library
2. Build the .jar file from source as described in the README:

```
mvn clean install -Dgpg.skip=true
```

3. Copy the resulting .jar file into your project and add it to your project's `build.xml` file. For example:

```
<classpath>
  <pathelement path="${classpath}"/>
  <pathelement location="lib/aws-cloudtrail-processing-library-1.6.1.jar"/>
</classpath>
```

Adding the library to an Apache Maven project

The CloudTrail Processing Library is available for Apache Maven. You can add it to your project by writing a single dependency in your project's `pom.xml` file.

To add the CloudTrail Processing Library to a Maven project

- Open your Maven project's `pom.xml` file and add the following dependency:

```
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>aws-cloudtrail-processing-library</artifactId>
    <version>1.6.1</version>
</dependency>
```

Adding the library to an Eclipse project

To add the CloudTrail Processing Library to an Eclipse project

1. Download or clone the CloudTrail Processing Library source code from GitHub:
   - https://github.com/aws/aws-cloudtrail-processing-library
2. Build the .jar file from source as described in the README:

```
mvn clean install -Dgpg.skip=true
```

3. Copy the built aws-cloudtrail-processing-library-1.6.1.jar to a directory in your project (typically `lib`).
4. Right-click your project's name in the Eclipse **Project Explorer**, choose **Build Path**, and then choose **Configure**
5. In the **Java Build Path** window, choose the **Libraries** tab.
6. Choose **Add JARs...** and navigate to the path where you copied aws-cloudtrail-processing-library-1.6.1.jar.
7. Choose **OK** to complete adding the `.jar` to your project.

Adding the library to an IntelliJ project

To add the CloudTrail Processing Library to an IntelliJ project

1. Download or clone the CloudTrail Processing Library source code from GitHub:
   - https://github.com/aws/aws-cloudtrail-processing-library
2. Build the .jar file from source as described in the README:

```
mvn clean install -Dgpg.skip=true
```

3. From **File**, choose **Project Structure**.
4. Choose **Modules** and then choose **Dependencies**.
5. Choose **+ JARS or Directories** and then go to the path where you built the `aws-cloudtrail-processing-library-1.6.1.jar`.
6. Choose **Apply** and then choose **OK** to complete adding the `.jar` to your project.

Configuring the CloudTrail Processing Library

You can configure the CloudTrail Processing Library by creating a classpath properties file that is loaded at runtime, or by creating a `ClientConfiguration` object and setting options manually.

Providing a properties file

You can write a classpath properties file that provides configuration options to your application. The following example file shows the options you can set:

```
# AWS access key. (Required)
accessKey = your_access_key

# AWS secret key. (Required)
secretKey = your_secret_key

# The SQS URL used to pull CloudTrail notification from. (Required)
sqsUrl = your_sqs_queue_url

# The SQS end point specific to a region.
sqsRegion = us-east-1

# A period of time during which Amazon SQS prevents other consuming components
# from receiving and processing that message.
visibilityTimeout = 60

# The S3 region to use.
s3Region = us-east-1

# Number of threads used to download S3 files in parallel. Callbacks can be
# invoked from any thread.
threadCount = 1

# The time allowed, in seconds, for threads to shut down after
# AWSCloudTrailEventProcessingExecutor.stop() is called. If they are still
# running beyond this time, they will be forcibly terminated.
threadTerminationDelaySeconds = 60

# The maximum number of AWSCloudTrailClientEvents sent to a single invocation
# of processEvents().
maxEventsPerEmit = 10
```

```
# Whether to include raw event information in CloudTrailDeliveryInfo.
enableRawEventInfo = false

# Whether to delete SQS message when the CloudTrail Processing Library is unable to process
 the notification.
deleteMessageUponFailure = false
```

The following parameters are required:

- `sqsUrl` – Provides the URL from which to pull your CloudTrail notifications. If you don't specify this value, the `AWSCloudTrailProcessingExecutor` throws an `IllegalStateException`.
- `accessKey` – A unique identifier for your account, such as AKIAIOSFODNN7EXAMPLE.
- `secretKey` – A unique identifier for your account, such as wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY.

The `accessKey` and `secretKey` parameters provide your Amazon credentials to the library so the library can access Amazon on your behalf.

Defaults for the other parameters are set by the library. For more information, see the Amazon CloudTrail Processing Library Reference.

Creating a ClientConfiguration

Instead of setting options in the classpath properties, you can provide options to the `AWSCloudTrailProcessingExecutor` by initializing and setting options on a `ClientConfiguration` object, as shown in the following example:

```
ClientConfiguration basicConfig = new ClientConfiguration(
    "http://sqs.us-east-1.amazonaws.com/123456789012/queue2",
    new DefaultAWSCredentialsProviderChain());

basicConfig.setEnableRawEventInfo(true);
basicConfig.setThreadCount(4);
basicConfig.setnEventsPerEmit(20);
```

Implementing the events processor

To process CloudTrail logs, you must implement an `EventsProcessor` that receives the CloudTrail log data. The following is an example implementation:

```
public class SampleEventsProcessor implements EventsProcessor {

    public void process(List<CloudTrailEvent> events) {
        int i = 0;
        for (CloudTrailEvent event : events) {
            System.out.println(String.format("Process event %d : %s", i++,
 event.getEventData()));
        }
    }
}
```

When implementing an `EventsProcessor`, you implement the `process()` callback that the `AWSCloudTrailProcessingExecutor` uses to send you CloudTrail events. Events are provided in a list of `CloudTrailClientEvent` objects.

The `CloudTrailClientEvent` object provides a `CloudTrailEvent` and `CloudTrailEventMetadata` that you can use to read the CloudTrail event and delivery information.

This simple example prints the event information for each event passed to `SampleEventsProcessor`. In your own implementation, you can process logs as you see fit. The `AWSCloudTrailProcessingExecutor` continues to send events to your `EventsProcessor` as long as it has events to send and is still running.

Instantiating and running the processing executor

After you write an `EventsProcessor` and set configuration values for the CloudTrail Processing Library (either in a properties file or by using the `ClientConfiguration` class), you can use these elements to initialize and use an `AWSCloudTrailProcessingExecutor`.

To use `AWSCloudTrailProcessingExecutor` to process CloudTrail events

1. Instantiate an `AWSCloudTrailProcessingExecutor.Builder` object. `Builder`'s constructor takes an `EventsProcessor` object and a classpath properties file name.
2. Call the `Builder`'s `build()` factory method to configure and obtain an `AWSCloudTrailProcessingExecutor` object.
3. Use the `AWSCloudTrailProcessingExecutor`'s `start()` and `stop()` methods to begin and end CloudTrail event processing.

```
public class SampleApp {
  public static void main(String[] args) throws InterruptedException {
    AWSCloudTrailProcessingExecutor executor = new
      AWSCloudTrailProcessingExecutor.Builder(new SampleEventsProcessor(),
        "/myproject/cloudtrailprocessing.properties").build();

    executor.start();
    Thread.sleep(24 * 60 * 60 * 1000); // let it run for a while (optional)
    executor.stop(); // optional
  }
}
```

Advanced topics

Topics

- Filtering the events to process (p. 331)
- Processing data events (p. 333)
- Reporting progress (p. 333)
- Handling errors (p. 334)

Filtering the events to process

By default, all logs in your Amazon SQS queue's S3 bucket and all events that they contain are sent to your `EventsProcessor`. The CloudTrail Processing Library provides optional interfaces that you can implement to filter the sources used to obtain CloudTrail logs and to filter the events that you are interested in processing.

`SourceFilter`

You can implement the `SourceFilter` interface to choose whether you want to process logs from a provided source. `SourceFilter` declares a single callback method, `filterSource()`, that receives a `CloudTrailSource` object. To keep events from a source from being processed, return `false` from `filterSource()`.

The CloudTrail Processing Library calls the `filterSource()` method after the library polls for logs on the Amazon SQS queue. This occurs before the library starts event filtering or processing for the logs.

The following is an example implementation:

```
public class SampleSourceFilter implements SourceFilter{
  private static final int MAX_RECEIVED_COUNT = 3;

  private static List<String> accountIDs ;
  static {
    accountIDs = new ArrayList<>();
    accountIDs.add("123456789012");
    accountIDs.add("234567890123");
  }

  @Override
  public boolean filterSource(CloudTrailSource source) throws CallbackException {
    source = (SQSBasedSource) source;
    Map<String, String> sourceAttributes = source.getSourceAttributes();

    String accountId = sourceAttributes.get(
      SourceAttributeKeys.ACCOUNT_ID.getAttributeKey());

    String receivedCount = sourceAttributes.get(
      SourceAttributeKeys.APPROXIMATE_RECEIVE_COUNT.getAttributeKey());

    int approximateReceivedCount = Integer.parseInt(receivedCount);

    return approximateReceivedCount <= MAX_RECEIVED_COUNT &&
 accountIDs.contains(accountId);
  }
}
```

If you don't provide your own `SourceFilter`, then `DefaultSourceFilter` is used, which allows all sources to be processed (it always returns `true`).

`EventFilter`

You can implement the `EventFilter` interface to choose whether a CloudTrail event is sent to your `EventsProcessor`. `EventFilter` declares a single callback method, `filterEvent()`, that receives a `CloudTrailEvent` object. To keep the event from being processed, return `false` from `filterEvent()`.

The CloudTrail Processing Library calls the `filterEvent()` method after the library polls for logs on the Amazon SQS queue and after source filtering. This occurs before the library starts event processing for the logs.

See the following example implementation:

```
public class SampleEventFilter implements EventFilter{

  private static final String EC2_EVENTS = "ec2.amazonaws.com";

  @Override
  public boolean filterEvent(CloudTrailClientEvent clientEvent) throws
 CallbackException {
    CloudTrailEvent event = clientEvent.getEvent();

    String eventSource = event.getEventSource();
    String eventName = event.getEventName();
```

```
    return eventSource.equals(EC2_EVENTS) && eventName.startsWith("Delete");
  }
}
```

If you don't provide your own `EventFilter`, then `DefaultEventFilter` is used, which allows all events to be processed (it always returns `true`).

Processing data events

When CloudTrail processes data events, it preserves numbers in their original format, whether that is an integer (`int`) or a `float` (a number that contains a decimal). In events that have integers in the fields of a data event, CloudTrail historically processed these numbers as floats. Currently, CloudTrail processes numbers in these fields by keeping their original format.

As a best practice, to avoid breaking your automations, be flexible in any code or automation that you are using to process or filter CloudTrail data events, and allow both `int` and `float` formatted numbers. For best results, use version 1.4.0 or higher of the CloudTrail Processing Library.

The following example snippet shows a `float` formatted number, `2.0`, for the `desiredCount` parameter in the `ResponseParameters` block of a data event.

```
"eventName": "CreateService",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "000.00.00.00",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "clientToken": "EXAMPLE",
        "cluster": "default",
        "desiredCount": 2.0
...
```

The following example snippet shows an `int` formatted number, 2, for the `desiredCount` parameter in the `ResponseParameters` block of a data event.

```
"eventName": "CreateService",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "000.00.00.00",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "clientToken": "EXAMPLE",
        "cluster": "default",
        "desiredCount": 2
...
```

Reporting progress

Implement the `ProgressReporter` interface to customize the reporting of CloudTrail Processing Library progress. `ProgressReporter` declares two methods: `reportStart()` and `reportEnd()`, which are called at the beginning and end of the following operations:

- Polling messages from Amazon SQS
- Parsing messages from Amazon SQS
- Processing an Amazon SQS source for CloudTrail logs
- Deleting messages from Amazon SQS
- Downloading a CloudTrail log file

- Processing a CloudTrail log file

Both methods receive a `ProgressStatus` object that contains information about the operation that was performed. The `progressState` member holds a member of the `ProgressState` enumeration that identifies the current operation. This member can contain additional information in the `progressInfo` member. Additionally, any object that you return from `reportStart()` is passed to `reportEnd()`, so you can provide contextual information such as the time when the event began processing.

The following is an example implementation that provides information about how long an operation took to complete:

```
public class SampleProgressReporter implements ProgressReporter {
  private static final Log logger =
    LogFactory.getLog(DefaultProgressReporter.class);

  @Override
  public Object reportStart(ProgressStatus status) {
    return new Date();
  }

  @Override
  public void reportEnd(ProgressStatus status, Object startDate) {
    System.out.println(status.getProgressState().toString() + " is " +
      status.getProgressInfo().isSuccess() + " , and latency is " +
      Math.abs(((Date) startDate).getTime()-new Date().getTime()) + "
      milliseconds.");
  }
}
```

If you don't implement your own `ProgressReporter`, then `DefaultExceptionHandler`, which prints the name of the state being run, is used instead.

Handling errors

The `ExceptionHandler` interface allows you to provide special handling when an exception occurs during log processing. `ExceptionHandler` declares a single callback method, `handleException()`, which receives a `ProcessingLibraryException` object with context about the exception that occurred.

You can use the passed-in `ProcessingLibraryException`'s `getStatus()` method to find out what operation was executed when the exception occurred and get additional information about the status of the operation. `ProcessingLibraryException` is derived from Java's standard `Exception` class, so you can also retrieve information about the exception by invoking any of the exception methods.

See the following example implementation:

```
public class SampleExceptionHandler implements ExceptionHandler{
  private static final Log logger =
    LogFactory.getLog(DefaultProgressReporter.class);

  @Override
  public void handleException(ProcessingLibraryException exception) {
    ProgressStatus status = exception.getStatus();
    ProgressState state = status.getProgressState();
    ProgressInfo info = status.getProgressInfo();

    System.err.println(String.format(
      "Exception. Progress State: %s. Progress Information: %s.", state, info));
  }
```

```
}
```

If you don't provide your own `ExceptionHandler`, then `DefaultExceptionHandler`, which prints a standard error message, is used instead.

Note
If the `deleteMessageUponFailure` parameter is `true`, the CloudTrail Processing Library does not distinguish general exceptions from processing errors and may delete queue messages.

1. For example, you use the `SourceFilter` to filter messages by timestamp.
2. However, you don't have the required permissions to access the S3 bucket that receives the CloudTrail log files. Because you don't have the required permissions, an `AmazonServiceException` is thrown. The CloudTrail Processing Library wraps this in a `CallBackException`.
3. The `DefaultExceptionHandler` logs this as an error, but does not identify the root cause, which is that you don't have the required permissions. The CloudTrail Processing Library considers this a processing error and deletes the message, even if the message includes a valid CloudTrail log file.

If you want to filter messages with `SourceFilter`, verify that your `ExceptionHandler` can distinguish service exceptions from processing errors.

Additional resources

For more information about the CloudTrail Processing Library, see the following:

- CloudTrail Processing Library GitHub project, which includes sample code that demonstrates how to implement a CloudTrail Processing Library application.
- CloudTrail Processing Library Java Package Documentation.

Settings

You can use the **Settings** page to review configured CloudTrail settings.

To access settings

1. Choose **Settings** in the left navigation pane of the CloudTrail console.
2. Review and update your settings as needed.

The following settings are available:

- Organization delegated administrator (p. 336)

Organization delegated administrator

When you use CloudTrail with an Amazon Organizations organization, you can assign any account within the organization to act as a CloudTrail delegated administrator to manage the organization's trails and event data stores on behalf of the organization. A delegated administrator is a member account in an organization that can perform the same administrative tasks in CloudTrail as the management account. Only the organization management account can assign a CloudTrail delegated administrator.

If you choose a delegated administrator, this member account has administrative permissions on all trails and event data stores in the organization. Adding a delegated administrator does not disrupt the management or operation of the organization's trails or event data stores.

The first time you add a delegated administrator in the CloudTrail console, or by using the Amazon CLI or CloudTrail API, CloudTrail checks whether the organization’s management account has a service-linked role. If the management account does not have a service-linked role, CloudTrail creates the service-linked role for the management account. For more information about service-linked roles, see Using service-linked roles for Amazon CloudTrail (p. 377).

> **Note**
> When you add a delegated administrator using the Amazon Organizations CLI or API operation, the service-linked role doesn't get created if it does not exist. The service-linked role is only created when you make a call from the management account directly to the CloudTrail service, such as when you add a delegated administrator or create an organization trail or event data store using the CloudTrail console, Amazon CLI or CloudTrail API.

Take note of the following factors that define how the delegated administrator operates in CloudTrail.

The management account remains the owner of any CloudTrail organization resources the delegated administrator creates.

The organization's management account remains the owner of any CloudTrail organization resources the delegated administrator creates, such as trails and event data stores. This provides continuity for the organization in the event the delegated administrator changes.

Removing a delegated administrator account does not delete any CloudTrail organization resources they created.

Organization trails and event data stores created by the delegated administrator are not deleted when you remove the delegated administrator, because the management account always serves as the owner of the CloudTrail organization resources regardless of whether they are created by the delegated administrator or the management account.

An organization can have a maximum of three CloudTrail delegated administrators.

You can have a maximum of three CloudTrail delegated administrators per organization. For more information about removing a delegated administrator, see Remove a CloudTrail delegated administrator (p. 338).

Topics

- Required permissions to assign a delegated administrator (p. 337)
- Add a CloudTrail delegated administrator (p. 337)
- Remove a CloudTrail delegated administrator (p. 338)

Required permissions to assign a delegated administrator

When assigning a CloudTrail delegated administrator, you must have the permissions to add and remove the delegated administrator in CloudTrail, as well as certain Amazon Organizations API actions and IAM permissions listed in the following policy statement.

You can add the following statement to the end of an IAM policy to grant these permissions:

```
{
    "Sid": "Permissions",
    "Effect": "Allow",
    "Action": [
        "cloudtrail:RegisterOrganizationDelegatedAdmin",
        "cloudtrail:DeregisterOrganizationDelegatedAdmin",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:DeregisterDelegatedAdministrator",
        "organizations:ListAWSServiceAccessForOrganization",
        "iam:CreateServiceLinkedRole",
        "iam:GetRole"
    ],
    "Resource": "*"
}
```

Add a CloudTrail delegated administrator

You can add a delegated administrator to manage an organization's CloudTrail resources, such as trails and event data stores.

You can add a CloudTrail delegated administrator for your Amazon organization using the Amazon CLI.

Before you add a delegated administrator, be sure they have an account in your organization and you are signed in with the management account for your organization. For information about how to create a new Amazon account for your organization, see Creating an Amazon account in your organization. For information about how to invite an existing Amazon account to your organization, see Inviting an Amazon account to join your organization.

Amazon CLI

The following example adds a CloudTrail delegated administrator.

```
aws cloudtrail register-organization-delegated-admin
  --member-account-id="memberAccountId"
```

This command produces no output if it's successful.

Remove a CloudTrail delegated administrator

You can remove a CloudTrail delegated administrator using the Amazon CLI.

Amazon CLI

The following command removes a CloudTrail delegated administrator.

```
aws cloudtrail deregister-organization-delegated-admin
  --delegated-admin-account-id="delegatedAdminAccountId"
```

This command produces no output if it's successful.

Security in Amazon CloudTrail

Cloud security at Amazon is the highest priority. As an Amazon customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between Amazon and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – Amazon is responsible for protecting the infrastructure that runs Amazon services in the Amazon Cloud. Amazon 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 Amazon compliance programs. To learn about the compliance programs that apply to Amazon CloudTrail, see Amazon Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the Amazon 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 CloudTrail. The following topics show you how to configure CloudTrail to meet your security and compliance objectives. You also learn how to use other Amazon services that help you to monitor and secure your CloudTrail resources.

Topics

- Data protection in Amazon CloudTrail (p. 339)
- Identity and Access Management for Amazon CloudTrail (p. 340)
- Compliance validation for Amazon CloudTrail (p. 380)
- Resilience in Amazon CloudTrail (p. 380)
- Infrastructure security in Amazon CloudTrail (p. 381)
- Cross-service confused deputy prevention (p. 381)
- Security best practices in Amazon CloudTrail (p. 382)
- Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386)

Data protection in Amazon CloudTrail

The Amazon shared responsibility model applies to data protection in Amazon CloudTrail. As described in this model, Amazon is responsible for protecting the global infrastructure that runs all of the Amazon Web Services Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the Amazon Web Services that you use. For more information about data privacy, see the Data Privacy FAQ.

For data protection purposes, we recommend that you protect Amazon Web Services account credentials and set up individual users with Amazon IAM Identity Center or Amazon 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 Amazon resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with Amazon CloudTrail.
- Use Amazon encryption solutions, along with all default security controls within Amazon Web Services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing Amazon through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with CloudTrail or other Amazon Web Services using the console, API, Amazon CLI, or Amazon 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.

By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption (SSE). You can also choose to encrypt your log files with an Amazon Key Management Service (Amazon KMS) key. You can store your log files in your bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.

The following security best practices also address data protection in CloudTrail:

- Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386)
- Amazon S3 bucket policy for CloudTrail (p. 363)
- Validating CloudTrail log file integrity (p. 307)
- Sharing CloudTrail log files between Amazon accounts (p. 297)

Because CloudTrail logs files are stored in a bucket or buckets in Amazon S3, you should also review the data protection information in the Amazon Simple Storage Service User Guide. For more information, see Protecting Data in Amazon S3.

Identity and Access Management for Amazon CloudTrail

Amazon Identity and Access Management (IAM) is an Amazon Web Service that helps an administrator securely control access to Amazon resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use CloudTrail resources. IAM is an Amazon Web Service that you can use with no additional charge.

Topics

- Audience (p. 341)
- Authenticating with identities (p. 341)
- Managing access using policies (p. 343)
- How Amazon CloudTrail works with IAM (p. 344)
- Identity-based policy examples for Amazon CloudTrail (p. 350)
- Amazon CloudTrail resource-based policy examples (p. 361)
- Amazon S3 bucket policy for CloudTrail (p. 363)
- Amazon S3 bucket policy for CloudTrail Lake query results (p. 368)
- Amazon SNS topic policy for CloudTrail (p. 369)

- Troubleshooting Amazon CloudTrail identity and access (p. 374)
- Using service-linked roles for Amazon CloudTrail (p. 377)
- Amazon managed policies for Amazon CloudTrail (p. 378)

Audience

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

Service user – If you use the CloudTrail service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more CloudTrail 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 CloudTrail, see Troubleshooting Amazon CloudTrail identity and access (p. 374).

Service administrator – If you're in charge of CloudTrail resources at your company, you probably have full access to CloudTrail. It's your job to determine which CloudTrail 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 CloudTrail, see How Amazon CloudTrail works with IAM (p. 344).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to CloudTrail. To view example CloudTrail identity-based policies that you can use in IAM, see Identity-based policy examples for Amazon CloudTrail (p. 350).

Authenticating with identities

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

If you access Amazon programmatically, Amazon 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 Amazon tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing Amazon 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, Amazon recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in Amazon in the *IAM User Guide*.

Amazon Web Services account root user

When you create an Amazon Web Services account, you begin with one sign-in identity that has complete access to all Amazon Web Services and resources in the account. This identity is called the Amazon Web Services 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 *Amazon Account Management Reference 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 Amazon Web Services by using temporary credentials.

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

IAM users and groups

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

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

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

IAM roles

An *IAM role* is an identity within your Amazon Web Services account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the Amazon Web Services Management Console by switching roles. You can assume a role by calling an Amazon CLI or Amazon API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*.
- **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 Amazon Web Services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some Amazon Web Services use features in other Amazon Web Services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in Amazon, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for Amazon CloudTrail in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more

information, see Creating a role to delegate permissions to an Amazon Web Service in the *IAM User Guide*.

- **Service-linked role** – A service-linked role is a type of service role that is linked to an Amazon Web Service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your Amazon Web Services 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 Amazon CLI or Amazon API requests. This is preferable to storing access keys within the EC2 instance. To assign an Amazon role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

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

Managing access using policies

You control access in Amazon by creating policies and attaching them to Amazon identities or resources. A policy is an object in Amazon that, when associated with an identity or resource, defines their permissions. Amazon 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 Amazon 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 Amazon 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 Amazon Web Services Management Console, the Amazon CLI, or the Amazon API.

Identity-based policies

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

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

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-

based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or Amazon Web Services.

Resource-based policies are inline policies that are located in that service. You can't use Amazon 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, Amazon 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

Amazon 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 Amazon Organizations. Amazon Organizations is a service for grouping and centrally managing multiple Amazon Web Services 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 Amazon Web Services account root user. For more information about Organizations and SCPs, see How SCPs work in the *Amazon 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 Amazon determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Amazon CloudTrail works with IAM

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

IAM features you can use with Amazon CloudTrail

IAM feature	CloudTrail support
Identity-based policies (p. 345)	Yes
Resource-based policies (p. 345)	Partial
Policy actions (p. 346)	Yes
Policy resources (p. 347)	Yes
Policy condition keys (service-specific) (p. 348)	No
ACLs (p. 348)	No
ABAC (tags in policies) (p. 349)	Yes
Temporary credentials (p. 349)	Yes
Principal permissions (p. 350)	Yes
Service roles (p. 350)	Yes
Service-linked roles (p. 350)	Yes

To get a high-level view of how CloudTrail and other Amazon services work with most IAM features, see Amazon services that work with IAM in the *IAM User Guide*.

Identity-based policies for CloudTrail

Supports identity-based policies	Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for CloudTrail

To view examples of CloudTrail identity-based policies, see Identity-based policy examples for Amazon CloudTrail (p. 350).

Resource-based policies within CloudTrail

Supports resource-based policies	Partial

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 Amazon Web 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 Amazon Web Services accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

CloudTrail supports resource-based policies on channels used for CloudTrail Lake integrations with event sources outside of Amazon. The resource-based policy for the channel defines which principal entities (accounts, users, roles, and federated users) can call `PutAuditEvents` on the channel to deliver events to the destination event data store. For more information about creating integrations with CloudTrail Lake, see Create an integration with an event source outside of Amazon (p. 180).

Examples

To view examples of CloudTrail resource-based policies, see Amazon CloudTrail resource-based policy examples (p. 361).

Policy actions for CloudTrail

Supports policy actions	Yes

Administrators can use Amazon 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 Amazon API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

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

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

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

```
cloudtrail
```

For example, to grant someone permission to list tags for a trail with the `ListTags` API operation, you include the `cloudtrail:ListTags` action in their policy. Policy statements must include either an `Action` or `NotAction` element. CloudTrail 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 as follows:

```
"Action": [
      "cloudtrail:AddTags",
```

```
    "cloudtrail:ListTags",
    "cloudtrail:RemoveTags
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Get`, include the following action:

```
"Action": "cloudtrail:Get*"
```

Policy resources for CloudTrail

Supports policy resources	Yes

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

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

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

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

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

In CloudTrail, there are three resource types: trails, event data stores, and channels. Each resource has a unique Amazon Resource Name (ARN) associated with it. In a policy, you use an ARN to identify the resource that the policy applies to. CloudTrail does not currently support other resource types, which are sometimes referred to as subresources.

The CloudTrail trail resource has the following ARN:

```
arn:${Partition}:cloudtrail:${Region}:${Account}:trail/{TrailName}
```

The CloudTrail event data store resource has the following ARN:

```
arn:${Partition}:cloudtrail:${Region}:${Account}:eventdatastore/{EventDataStoreId}
```

The CloudTrail channel resource has the following ARN:

```
arn:${Partition}:cloudtrail:${Region}:${Account}:channel/{ChannelId}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and Amazon Service Namespaces.

For example, for an Amazon Web Services account with the ID *123456789012*, to specify a trail named *My-Trail* that exists in the US East (Ohio) Region in your statement, use the following ARN:

```
"Resource": "arn:aws:cloudtrail:us-east-2:123456789012:trail/My-Trail"
```

To specify all trails that belong to a specific account in that Amazon Web Services Region, use the wildcard (*):

```
"Resource": "arn:aws:cloudtrail:us-east-2:123456789012:trail/*"
```

Some CloudTrail actions, such as those for creating resources, can't be performed on a specific resource. In those cases, you must use the wildcard (*).

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

Many CloudTrail API actions involve multiple resources. For example, `CreateTrail` requires an Amazon S3 bucket for storing log files, so a user must have permissions to write to the bucket. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "resource1",
      "resource2"
```

Policy condition keys for CloudTrail

Supports service-specific policy condition keys	No

Administrators can use Amazon 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, Amazon evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, Amazon 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*.

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

CloudTrail doesn't define its own condition keys, but it supports using some global condition keys. To see all Amazon global condition keys, see Amazon Global Condition Context Keys in the *IAM User Guide*.

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

ACLs in CloudTrail

Supports ACLs	No

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

ABAC with CloudTrail

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In Amazon, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many Amazon 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 What is ABAC? 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*.

Although you can attach tags to CloudTrail resources, CloudTrail only supports controlling access to CloudTrail Lake (p. 156) event data stores based on tags. You cannot control access to trails based on tags.

You can attach tags to CloudTrail resources or pass tags in a request to CloudTrail. For more information about tagging CloudTrail resources, see Creating a trail (p. 78) and Creating, updating, and managing trails with the Amazon Command Line Interface (p. 97).

Using temporary credentials with CloudTrail

Supports temporary credentials	Yes

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

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

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

Principal permissions for CloudTrail

Supports principal permissions	Yes

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

Service roles for CloudTrail

Supports service roles	Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an Amazon Web Service in the *IAM User Guide*.

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

Service-linked roles for CloudTrail

Supports service-linked roles	Yes

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

CloudTrail supports a service-linked role for integration with Amazon Organizations. This role is required for the creation of an organization trail or event data store. Organization trails and event data stores log events for all Amazon Web Services accounts in an organization. For more information about creating or managing CloudTrail service-linked roles, see Using service-linked roles for Amazon CloudTrail (p. 377).

Identity-based policy examples for Amazon CloudTrail

By default, users and roles don't have permission to create or modify CloudTrail resources. They also can't perform tasks by using the Amazon Web Services Management Console, Amazon Command Line Interface (Amazon CLI), or Amazon API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

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

For details about actions and resource types defined by CloudTrail, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for Amazon CloudTrail in the *Service Authorization Reference*.

Topics

- Policy best practices (p. 351)
- Example: Allowing and denying actions for a specified trail (p. 351)
- Examples: Creating and applying policies for actions on specific trails (p. 352)
- Examples: Denying access to create or delete event data stores based on tags (p. 354)
- Using the CloudTrail console (p. 355)
- Allow users to view their own permissions (p. 356)
- Granting custom permissions for CloudTrail users (p. 357)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete CloudTrail resources in your account. These actions can incur costs for your Amazon Web Services account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with Amazon managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *Amazon managed policies* that grant permissions for many common use cases. They are available in your Amazon Web Services account. We recommend that you reduce permissions further by defining Amazon customer managed policies that are specific to your use cases. For more information, see Amazon managed policies or Amazon 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 Amazon Web Service, such as Amazon 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 IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your Amazon Web Services account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

CloudTrail doesn't have service-specific context keys that you can use in the `Condition` element of policy statements.

Example: Allowing and denying actions for a specified trail

The following example demonstrates a policy that allows users with the policy to view the status and configuration of a trail and start and stop logging for a trail named *`My-First-Trail`*. This trail was created in the US East (Ohio) Region (its home Region) in the Amazon Web Services account with the ID *`123456789012`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "cloudtrail:StartLogging",
              "cloudtrail:StopLogging",
              "cloudtrail:GetTrail",
              "cloudtrail:GetTrailStatus",
              "cloudtrail:GetEventSelectors"
          ],
          "Resource": [
              "arn:aws:cloudtrail:us-east-2:123456789012:trail/My-First-Trail"
          ]
      }
  ]
}
```

The following example demonstrates a policy that explicitly denies CloudTrail actions for any trail not named *My-First-Trail*.

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Deny",
          "Action": [
              "cloudtrail:*"
          ],
          "NotResource": [
              "arn:aws:cloudtrail:us-east-2:123456789012:trail/My-First-Trail"
          ]
      }
  ]
}
```

Examples: Creating and applying policies for actions on specific trails

You can use permissions and policies to control a user's ability to perform specific actions on CloudTrail trails.

For example, you don't want users of your company's developer group to start or stop logging on a specific trail. However, you might want to grant them permission to perform the `DescribeTrails` and `GetTrailStatus` actions on the trail. You want the users of the developer group to perform the `StartLogging` or `StopLogging` actions on trails that they manage.

You can create two policy statements and attach them to the developer group you create in IAM. For more information about groups in IAM, see IAM Groups in the *IAM User Guide*.

In the first policy, you deny the `StartLogging` and `StopLogging` actions for the trail ARN that you specify. In the following example, the trail ARN is `arn:aws:cloudtrail:us-east-2:123456789012:trail/Example-Trail`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1446057698000",
```

```
            "Effect": "Deny",
            "Action": [
                "cloudtrail:StartLogging",
                "cloudtrail:StopLogging"
            ],
            "Resource": [
                "arn:aws:cloudtrail:us-east-2:123456789012:trail/Example-Trail"
            ]
        }
    ]
}
```

In the second policy, the `DescribeTrails` and `GetTrailStatus` actions are allowed on all CloudTrail resources:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1446072643000",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:DescribeTrails",
                "cloudtrail:GetTrail",
                "cloudtrail:GetTrailStatus"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

If a user of the developer group tries to start or stop logging on the trail that you specified in the first policy, that user gets an access denied exception. Users of the developer group can start and stop logging on trails that they create and manage.

The following examples show that the configured developer group in an Amazon CLI profile named `devgroup`. First, a user of `devgroup` runs the `describe-trails` command.

```
$ aws --profile devgroup cloudtrail describe-trails
```

The command complete successfully with the following output:

```
{
    "trailList": [
        {
            "IncludeGlobalServiceEvents": true,
            "Name": "Default",
            "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/Example-Trail",
            "IsMultiRegionTrail": false,
            "S3BucketName": "myS3bucket ",
            "HomeRegion": "us-east-2"
        }
    ]
}
```

The user then runs the `get-trail-status` command on the trail that you specified in the first policy.

```
$ aws --profile devgroup cloudtrail get-trail-status --name Example-Trail
```

The command complete successfully with the following output:

```
{
    "LatestDeliveryTime": 1449517556.256,
    "LatestDeliveryAttemptTime": "2015-12-07T19:45:56Z",
    "LatestNotificationAttemptSucceeded": "",
    "LatestDeliveryAttemptSucceeded": "2015-12-07T19:45:56Z",
    "IsLogging": true,
    "TimeLoggingStarted": "2015-12-07T19:36:27Z",
    "StartLoggingTime": 1449516987.685,
    "StopLoggingTime": 1449516977.332,
    "LatestNotificationAttemptTime": "",
    "TimeLoggingStopped": "2015-12-07T19:36:17Z"
}
```

Next, a user in the `devgroup` group runs the `stop-logging` command on the same trail.

```
$ aws --profile devgroup cloudtrail stop-logging --name Example-Trail
```

The command returns an access denied exception, such as the following:

```
A client error (AccessDeniedException) occurred when calling the StopLogging operation:
 Unknown
```

The user runs the `start-logging` command on the same trail.

```
$ aws --profile devgroup cloudtrail start-logging --name Example-Trail
```

Again the command returns an access denied exception, such as the following:

```
A client error (AccessDeniedException) occurred when calling the StartLogging operation:
 Unknown
```

Examples: Denying access to create or delete event data stores based on tags

In the following policy example, permission to create an event data store with `CreateEventDataStore` is denied if at least one of the following conditions aren't met:

- The event data store doesn't have a tag key of `stage` applied to itself
- The value of the stage tag isn't `alpha`, `beta`, `gamma`, or `prod`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "cloudtrail:CreateEventDataStore",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/stage": "true"
                }
            }
        },
        {
```

```
            "Effect": "Deny",
            "Action": "cloudtrail:CreateEventDataStore",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "aws:RequestTag/stage": [
                        "alpha",
                        "beta",
                        "gamma",
                        "prod"
                    ]
                }
            }
        }
    ]
}
```

In the following policy example, permission to delete an event data store with `DeleteEventDataStore` is denied is if the event data store has a `stage` tag with a value of `prod`. A policy like this one can help protect an event data store from accidental deletion.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "cloudtrail:DeleteEventDataStore",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/stage": "prod"
                }
            }
        }
    ]
}
```

Using the CloudTrail console

To access the Amazon CloudTrail console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the CloudTrail resources in your Amazon Web Services 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 Amazon CLI or the Amazon API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

Granting permissions for CloudTrail administration

To allow IAM roles or users to administer a CloudTrail resource, such as a trail, event data store, or channel, you must grant explicit permissions to perform the actions associated with CloudTrail tasks. In most situations, you can use an Amazon managed policy that contains predefined permissions.

> **Note**
> The permissions you grant to users to perform CloudTrail administration tasks aren't the same as the permissions that CloudTrail requires to deliver log files to Amazon S3 buckets or send notifications to Amazon SNS topics. For more information about those permissions, see Amazon S3 bucket policy for CloudTrail (p. 363).
> If you configure integration with Amazon CloudWatch Logs, CloudTrail also requires a role that it can assume to deliver events to an Amazon CloudWatch Logs log group. You must create the

role that CloudTrail uses. For more information, see Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console (p. 360) and Sending events to CloudWatch Logs (p. 277).

The following Amazon managed policies are available for CloudTrail:

- **AWSCloudTrail_FullAccess** – This policy provides full access to CloudTrail actions on CloudTrail resources, such as trails, event data stores, and channels. This policy provides the required permissions to create, update, and delete CloudTrail trails, event data stores, and channels.

  This policy also provides permissions to manage the Amazon S3 bucket, the log group for CloudWatch Logs, and an Amazon SNS topic for a trail. However, the `AWSCloudTrail_FullAccess` managed policy doesn't provide permissions to delete the Amazon S3 bucket, the log group for CloudWatch Logs, or an Amazon SNS topic. For information about managed policies for other Amazon services, see the *Amazon Managed Policy Reference Guide*.

  Note
  The **AWSCloudTrail_FullAccess** policy isn't intended to be shared broadly across your Amazon Web Services account. Users with this role can turn off or reconfigure the most sensitive and important auditing functions in their Amazon Web Services accounts. For this reason, you must only apply this policy to account administrators. You must closely control and monitor use of this policy.
- **AWSCloudTrail_ReadOnlyAccess** – This policy grants permissions to view the CloudTrail console, including recent events and event history. This policy also allows you to view existing trails, event data stores, and channels. Roles and users with this policy can download the event history, but they can't create or update trails, event data stores, or channels.

To provide access, add permissions to your users, groups, or roles:

- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Creating a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Creating a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

Additional resources

To learn more about using IAM to give identities, such as users and roles, access to resources in your account, see Getting set up with IAM and Access management for Amazon resources in the *IAM User Guide*.

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

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 Amazon CLI or Amazon API.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws-cn: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": "*"
        }
    ]
}
```

Granting custom permissions for CloudTrail users

CloudTrail policies grant permissions to users who work with CloudTrail. If you need to grant different permissions to users, you can attach a CloudTrail policy to an IAM group or to a user. You can edit the policy to include or exclude specific permissions. You can also create your own custom policy. Policies are JSON documents that define the actions a user is allowed to perform and the resources that the user is allowed to perform those actions on. For specific examples, see Example: Allowing and denying actions for a specified trail (p. 351) and Examples: Creating and applying policies for actions on specific trails (p. 352).

Contents

- Read-only access (p. 357)
- Full access (p. 358)
- Granting permission to view Amazon Config information on the CloudTrail console (p. 360)
- Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console (p. 360)
- Additional information (p. 361)

Read-only access

The following example shows a policy that grants read-only access to CloudTrail trails. This is equivalent to the managed policy **AWSCloudTrail_ReadOnlyAccess**. It grants users permission to see trail information, but not to create or update trails.

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

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:Get*",
                "cloudtrail:Describe*",
                "cloudtrail:List*",
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

In the policy statements, the `Effect` element specifies whether the actions are allowed or denied. The `Action` element lists the specific actions that the user is allowed to perform. The `Resource` element lists the Amazon resources the user is allowed to perform those actions on. For policies that control access to CloudTrail actions, the `Resource` element is usually set to `*`, a wildcard that means "all resources."

The values in the `Action` element correspond to the APIs that the services support. The actions are preceded by `cloudtrail:` to indicate that they refer to CloudTrail actions. You can use the `*` wildcard character in the `Action` element , such as in the following examples:

- `"Action": ["cloudtrail:*Logging"]`

  This allows all CloudTrail actions that end with "Logging" (`StartLogging`, `StopLogging`).
- `"Action": ["cloudtrail:*"]`

  This allows all CloudTrail actions, but not actions for other Amazon services.
- `"Action": ["*"]`

  This allows all Amazon actions. This permission is suitable for a user who acts as an Amazon administrator for your account.

The read-only policy doesn't grant user permission for the `CreateTrail`, `UpdateTrail`, `StartLogging`, and `StopLogging` actions. Users with this policy are not allowed to create trails, update trails, or turn logging on and off. For the list of CloudTrail actions, see the Amazon CloudTrail API Reference.

Full access

The following example shows a policy that grants full access to CloudTrail. This is equivalent to the managed policy **AWSCloudTrail_FullAccess**. It grants users the permission to perform all CloudTrail actions. It also lets users log data events in Amazon S3 and Amazon Lambda, manage files in Amazon S3 buckets, manage how CloudWatch Logs monitors CloudTrail log events, and manage Amazon SNS topics in the account that the user is associated with.

> **Important**
> The **AWSCloudTrail_FullAccess** policy or equivalent permissions are not intended to be shared broadly across your Amazon account. Users with this role or equivalent access have the ability to disable or reconfigure the most sensitive and important auditing functions in their Amazon accounts. For this reason, this policy should be applied only to account administrators, and use of this policy should be closely controlled and monitored.

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

```
            "Action": [
                "sns:AddPermission",
                "sns:CreateTopic",
                "sns:SetTopicAttributes",
                "sns:GetTopicAttributes"
            ],
            "Resource": [
                "arn:aws:sns:*:*:aws-cloudtrail-logs*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy"
            ],
            "Resource": [
                "arn:aws:s3:::aws-cloudtrail-logs*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudtrail:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:aws-cloudtrail-logs*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "iam:GetRolePolicy",
                "iam:GetUser"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "iam:PassedToService": "cloudtrail.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateKey",
                "kms:CreateAlias",
                "kms:ListKeys",
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:ListFunctions"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:ListGlobalTables",
                "dynamodb:ListTables"
            ],
            "Resource": "*"
        }
    ]
}
```

Granting permission to view Amazon Config information on the CloudTrail console

You can view event information on the CloudTrail console, including resources that are related to that event. For these resources, you can choose the Amazon Config icon to view the timeline for that resource in the Amazon Config console. Attach this policy to your users to grant them read-only Amazon Config access. The policy doesn't grant them permission to change settings in Amazon Config.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "config:Get*",
            "config:Describe*",
            "config:List*"
        ],
        "Resource": "*"
    }]
}
```

For more information, see Viewing resources referenced with Amazon Config (p. 55).

Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console

You can view and configure delivery of events to CloudWatch Logs in the CloudTrail console if you have sufficient permissions. These are permissions that may be beyond those granted for CloudTrail

administrators. Attach this policy to administrators who will configure and manage CloudTrail integration with CloudWatch Logs. The policy doesn't grant them permissions in CloudTrail or in CloudWatch Logs directly, but instead grants the permissions required to create and configure the role CloudTrail will assume to successfully deliver events to your CloudWatch Logs group.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam:CreateRole",
            "iam:PutRolePolicy",
            "iam:AttachRolePolicy",
            "iam:ListRoles",
            "iam:GetRolePolicy",
            "iam:GetUser"
        ],
        "Resource": "*"
    }]
}
```

For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).

Additional information

To learn more about using IAM to give identities, such as users and roles, access to resources in your account, see Getting started and Access management for Amazon resources in the *IAM User Guide*.

Amazon CloudTrail resource-based policy examples

CloudTrail supports resource-based permissions policies for CloudTrail channels used for CloudTrail Lake integrations. For more information about creating integrations with CloudTrail Lake, see Create an integration with an event source outside of Amazon (p. 180).

The information required for the policy is determined by the integration type.

- For a direction integration, CloudTrail requires the policy to contain the partner's Amazon Web Services account IDs, and requires you to enter the unique external ID provided by the partner. CloudTrail automatically adds the partner's Amazon Web Services account IDs to the resource policy when you create an integration using the CloudTrail console. Refer to the partner's documentation to learn how to get the Amazon Web Services account numbers required for the policy.
- For a solution integration, you must specify at least one Amazon Web Services account ID as principal, and can optionally enter an external ID to prevent against confused deputy.

The following are requirements for the resource-based policy:

- The resource ARN defined in the policy must match the channel ARN the policy is attached to.
- The policy contains only one action: `cloudtrail-data:PutAuditEvents`
- The policy contains at least one statement. The policy can have a maximum of 20 statements.
- Each statement contains at least one principal. A statement can have a maximum of 50 principals.

The channel owner can call the `PutAuditEvents` API on the channel unless the policy denies the owner access to the resource.

Topics

- Example: Providing channel access to principals (p. 362)
- Example: Using an external ID to prevent against confused deputy (p. 362)

Example: Providing channel access to principals

The following example grants permissions to the principals with the ARNs `arn:aws:iam::111122223333:root`, `arn:aws:iam::444455556666:root`, and `arn:aws:iam::123456789012:root` to call the PutAuditEvents API on the CloudTrail channel with the ARN `arn:aws:cloudtrail:us-east-1:777788889999:channel/EXAMPLE-80b5-40a7-ae65-6e099392355b`.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Sid": "ChannelPolicy",
            "Effect": "Allow",
            "Principal":
            {
                "AWS":
                [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root",
                    "arn:aws:iam::123456789012:root"
                ]
            },
            "Action": "cloudtrail-data:PutAuditEvents",
            "Resource": "arn:aws:cloudtrail:us-east-1:777788889999:channel/
EXAMPLE-80b5-40a7-ae65-6e099392355b"
        }
    ]
}
```

Example: Using an external ID to prevent against confused deputy

The following example uses an external ID to address and prevent against confused deputy. The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action.

The integration partner creates the external ID to use in the policy. Then, it provides the external ID to you as part of creating the integration. The value can be any unique string, such as a passphrase or account number.

The example grants permissions to the principals with the ARNs `arn:aws:iam::111122223333:root`, `arn:aws:iam::444455556666:root`, and `arn:aws:iam::123456789012:root` to call the PutAuditEvents API on the CloudTrail channel resource if the call to the `PutAuditEvents` API includes the external ID value defined in the policy.

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

```
            "Effect": "Allow",
            "Principal":
            {
                "AWS":
                [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root",
                    "arn:aws:iam::123456789012:root"
                ]
            },
            "Action": "cloudtrail-data:PutAuditEvents",
            "Resource": "arn:aws:cloudtrail:us-east-1:777788889999:channel/
EXAMPLE-80b5-40a7-ae65-6e099392355b",
            "Condition":
            {
                "StringEquals":
                {
                    "cloudtrail:ExternalId": "uniquePartnerExternalID"
                }
            }
        }
    ]
}
```

Amazon S3 bucket policy for CloudTrail

By default, Amazon S3 buckets and objects are private. Only the resource owner (the Amazon account that created the bucket) can access the bucket and objects it contains. The resource owner can grant access permissions to other resources and users by writing an access policy.

To create or modify an Amazon S3 bucket to receive log files for an organization trail, you must change the bucket policy. For more information, see Creating a trail for an organization with the Amazon Command Line Interface (p. 135).

To deliver log files to an S3 bucket, CloudTrail must have the required permissions, and it cannot be configured as a Requester Pays bucket.

CloudTrail adds the following fields in the policy for you:

- The allowed SIDs
- The bucket name
- The service principal name for CloudTrail
- The name of the folder where the log files are stored, including the bucket name, a prefix (if you specified one), and your Amazon account ID

As a security best practice, add an `aws:SourceArn` condition key to the Amazon S3 bucket policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail writes to the S3 bucket only for a specific trail or trails. The value of `aws:SourceArn` is always the ARN of the trail (or array of trail ARNs) that is using the bucket to store logs. Be sure to add the `aws:SourceArn` condition key to S3 bucket policies for existing trails.

The following policy allows CloudTrail to write log files to the bucket from supported Amazon Web Services Regions. Replace *myBucketName*, *[optionalPrefix]/*, *myAccountID*, *region*, and *trailName* with the appropriate values for your configuration.

S3 bucket policy

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailAclCheck20150319",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws-cn:s3:::myBucketName",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailWrite20150319",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws-cn:s3:::myBucketName/[optionalPrefix]/AWSLogs/myAccountID/
*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:trail/trailName"
                }
            }
        }
    ]
}
```

For more information about Amazon Web Services Regions, see CloudTrail supported Regions (p. 15).

Note
In the China (Beijing) Region, use the service principal name: `"cloudtrail.amazonaws.com"`.

Contents

- Specifying an existing bucket for CloudTrail log delivery (p. 364)
- Receiving log files from other accounts (p. 365)
- Create or update an Amazon S3 bucket to use to store the log files for an organization trail (p. 135)
- Troubleshooting the Amazon S3 bucket policy (p. 366)
    - Common Amazon S3 policy configuration errors (p. 366)
    - Changing a prefix for an existing bucket (p. 367)
- Additional resources (p. 368)

Specifying an existing bucket for CloudTrail log delivery

If you specified an existing S3 bucket as the storage location for log file delivery, you must attach a policy to the bucket that allows CloudTrail to write to the bucket.

Note
As a best practice, use a dedicated S3 bucket for CloudTrail logs.

To add the required CloudTrail policy to an Amazon S3 bucket

1. Open the Amazon S3 console at https://console.amazonaws.cn/s3/.
2. Choose the bucket where you want CloudTrail to deliver your log files, and then choose **Permissions**.

3. Choose **Edit**.
4. Copy the S3 bucket policy (p. 363) to the **Bucket Policy Editor** window. Replace the placeholders in italics with the names of your bucket, prefix, and account number. If you specified a prefix when you created your trail, include it here. The prefix is an optional addition to the S3 object key that creates a folder-like organization in your bucket.

   Note
   If the existing bucket already has one or more policies attached, add the statements for CloudTrail access to that policy or policies. Evaluate the resulting set of permissions to be sure that they are appropriate for the users who will access the bucket.

Receiving log files from other accounts

You can configure CloudTrail to deliver log files from multiple Amazon accounts to a single S3 bucket. For more information, see Receiving CloudTrail log files from multiple accounts (p. 290).

Create or update an Amazon S3 bucket to use to store the log files for an organization trail

You must specify an Amazon S3 bucket to receive the log files for an organization trail. This bucket must have a policy that allows CloudTrail to put the log files for the organization into the bucket.

The following is an example policy for an Amazon S3 bucket named *myOrganizationBucket*. This bucket is in an Amazon account with the ID *111111111111*, which is the management account for an organization with the ID *o-exampleorgid* that allows logging for an organization trail. It also allows logging for the *111111111111* account in the event that the trail is changed from an organization trail to a trail for that account only. Replace *myOrganizationBucket*, *111111111111*, *region*, and *trailName* with the appropriate values for your configuration.

The example policy includes an `aws:SourceArn` condition key for the Amazon S3 bucket policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail writes to the S3 bucket only for a specific trail or trails. In an organization trail, the value of `aws:SourceArn` must be a trail ARN that is owned by the management account, and uses the management account ID.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailAclCheck20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::myOrganizationBucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
```

```
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::myOrganizationBucket/AWSLogs/111111111111/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailOrganizationWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cloudtrail.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::myOrganizationBucket/AWSLogs/o-exampleorgid/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:111111111111:trail/trailName"
                }
            }
        }
    ]
}
```

This example policy does not allow any users from member accounts to access the log files created for the organization. By default, organization log files are accessible only to the management account. For information about how to allow read access to the Amazon S3 bucket for IAM users in member accounts, see Sharing CloudTrail log files between Amazon accounts (p. 297).

Troubleshooting the Amazon S3 bucket policy

The following sections describe how to troubleshoot the S3 bucket policy.

Common Amazon S3 policy configuration errors

When you create a new bucket as part of creating or updating a trail, CloudTrail attaches the required permissions to your bucket. The bucket policy uses the service principal name, `"cloudtrail.amazonaws.com"`, which allows CloudTrail to deliver logs for all Regions.

If CloudTrail is not delivering logs for a Region, it's possible that your bucket has an older policy that specifies CloudTrail account IDs for each Region. This policy gives CloudTrail permission to deliver logs only for the Regions specified.

As a best practice, update the policy to use a permission with the CloudTrail service principal. To do this, replace the account ID ARNs with the service principal name: `"cloudtrail.amazonaws.com"`. This gives CloudTrail permission to deliver logs for current and new Regions. As a security best practice, add an `aws:SourceArn` or `aws:SourceAccount` condition key to the Amazon S3 bucket policy. This helps prevent unauthorized account access to your S3 bucket. If you have existing trails, be sure to add one or more condition keys. The following example shows a recommended policy configuration. Replace *myBucketName*, *[optionalPrefix]/*, *myAccountID*, *region*, and *trailName* with the appropriate values for your configuration.

Example Example bucket policy with service principal name

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailAclCheck20150319",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws-cn:s3:::myBucketName",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn":
 "arn:aws:cloudtrail:region:myAccountID:trail/trailName"
                }
            }
        },
        {
            "Sid": "AWSCloudTrailWrite20150319",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws-cn:s3:::myBucketName/[optionalPrefix]/AWSLogs/myAccountID/
*",
            "Condition": {"StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceArn": "arn:aws:cloudtrail:region:myAccountID:trail/trailName"
                }
            }
        }
    ]
}
```

Changing a prefix for an existing bucket

If you try to add, modify, or remove a log file prefix for an S3 bucket that receives logs from a trail, you might see the error: **There is a problem with the bucket policy**. A bucket policy with an incorrect prefix can prevent your trail from delivering logs to the bucket. To resolve this issue, use the Amazon S3 console to update the prefix in the bucket policy, and then use the CloudTrail console to specify the same prefix for the bucket in the trail.

To update the log file prefix for an Amazon S3 bucket

1. Open the Amazon S3 console at https://console.amazonaws.cn/s3/.
2. Choose the bucket for which you want to modify the prefix, and then choose **Permissions**.
3. Choose **Edit**.
4. In the bucket policy, under the `s3:PutObject` action, edit the `Resource` entry to add, modify, or remove the log file *`prefix/`* as needed.

   ```
   "Action": "s3:PutObject",
         "Resource": "arn:aws-cn:s3:::myBucketName/prefix/AWSLogs/myAccountID/*",
   ```

5. Choose **Save**.
6. Open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
7. Choose your trail and for **Storage location**, click the pencil icon to edit the settings for your bucket.
8. For **S3 bucket**, choose the bucket with the prefix you are changing.
9. For **Log file prefix**, update the prefix to match the prefix that you entered in the bucket policy.

10. Choose **Save**.

Additional resources

For more information about S3 buckets and policies, see the Amazon Simple Storage Service User Guide.

Amazon S3 bucket policy for CloudTrail Lake query results

By default, Amazon S3 buckets and objects are private. Only the resource owner (the Amazon account that created the bucket) can access the bucket and objects it contains. The resource owner can grant access permissions to other resources and users by writing an access policy.

To deliver CloudTrail Lake query results to an S3 bucket, CloudTrail must have the required permissions, and it cannot be configured as a Requester Pays bucket.

CloudTrail adds the following fields in the policy for you:

- The allowed SIDs
- The bucket name
- The service principal name for CloudTrail

As a security best practice, add an `aws:SourceArn` condition key to the Amazon S3 bucket policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail writes to the S3 bucket only for the event data store.

The following policy allows CloudTrail to deliver query results to the bucket from supported Amazon Web Services Regions. Replace *myBucketName*, *myAccountID*, and *myQueryRunningRegion* with the appropriate values for your configuration. The *myAccountID* is the Amazon account ID used for CloudTrail, which may not be the same as the Amazon account ID for the S3 bucket.

> **Note**
> If this is an organization event data store, you must use the Amazon account ID for the management account.

S3 bucket policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailLake1",
            "Effect": "Allow",
            "Principal": {"Service": "cloudtrail.amazonaws.com"},
            "Action": [
                "s3:PutObject*",
                "s3:Abort*"
            ],
            "Resource": [
                "arn:aws:s3:::myBucketName",
                "arn:aws:s3:::myBucketName/*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:sourceAccount": "myAccountID",
                    "aws:sourceArn":
 "arn:aws:cloudtrail:myQueryRunningRegion:myAccountID:eventdatastore/*"
```

```
                }
            }
        },
        {
            "Sid": "AWSCloudTrailLake2",
            "Effect": "Allow",
            "Principal": {"Service":"cloudtrail.amazonaws.com"},
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::myBucketName",
            "Condition": {
                "StringLike": {
                    "aws:sourceAccount": "myAccountID",
                    "aws:sourceArn":
 "arn:aws:cloudtrail:myQueryRunningRegion:myAccountID:eventdatastore/*"
                }
            }
        }
    ]
}
```

Contents

- Specifying an existing bucket for CloudTrail Lake query results (p. 369)
- Additional resources (p. 369)

Specifying an existing bucket for CloudTrail Lake query results

If you specified an existing S3 bucket as the storage location for CloudTrail Lake query results delivery, you must attach a policy to the bucket that allows CloudTrail to deliver the query results to the bucket.

> **Note**
> As a best practice, use a dedicated S3 bucket for CloudTrail Lake query results.

To add the required CloudTrail policy to an Amazon S3 bucket

1. Open the Amazon S3 console at https://console.amazonaws.cn/s3/.
2. Choose the bucket where you want CloudTrail to deliver your Lake query results, and then choose **Permissions**.
3. Choose **Edit**.
4. Copy the S3 bucket policy for query results (p. 368) to the **Bucket Policy Editor** window. Replace the placeholders in italics with the names of your bucket, Region, and account ID.

   > **Note**
   > If the existing bucket already has one or more policies attached, add the statements for CloudTrail access to that policy or policies. Evaluate the resulting set of permissions to be sure that they are appropriate for the users who access the bucket.

Additional resources

For more information about S3 buckets and policies, see Using bucket policies in the *Amazon Simple Storage Service User Guide*.

Amazon SNS topic policy for CloudTrail

To send notifications to an SNS topic, CloudTrail must have the required permissions. CloudTrail automatically attaches the required permissions to the topic when you create an Amazon SNS topic as part of creating or updating a trail in the CloudTrail console.

Important
As a security best practice, to restrict access to your SNS topic, we strongly recommend that after you create or update a trail to send SNS notifications, you manually edit the IAM policy that is attached to the SNS topic to add condition keys. For more information, see the section called "Security best practice for SNS topic policy" (p. 371) in this topic.

CloudTrail adds the following statement to the policy for you with the following fields:

- The allowed SIDs.
- The service principal name for CloudTrail.
- The SNS topic, including Region, account ID, and topic name.

The following policy allows CloudTrail to send notifications about log file delivery from supported Regions. For more information, see CloudTrail supported Regions (p. 15). This is the default policy that is attached to a new or existing SNS topic policy when you create or update a trail, and choose to enable SNS notifications.

SNS topic policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSCloudTrailSNSPolicy20131101",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:region:SNSTopicOwnerAccountId:SNSTopicName"
        }
    ]
}
```

To use an Amazon KMS-encrypted Amazon SNS topic to send notifications, you must also enable compatibility between the event source (CloudTrail) and the encrypted topic by adding the following statement to the policy of the Amazon KMS key.

KMS key policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information, see Enable Compatibility between Event Sources from Amazon Services and Encrypted Topics.

Contents

- Security best practice for SNS topic policy (p. 371)
- Specifying an existing topic for sending notifications (p. 372)
- Troubleshooting the SNS topic policy (p. 373)
    - Common SNS policy configuration errors (p. 373)
- Additional resources (p. 374)

Security best practice for SNS topic policy

By default, the IAM policy statement that CloudTrail attaches to your Amazon SNS topic allows the CloudTrail service principal to publish to an SNS topic, identified by an ARN. To help prevent an attacker from gaining access to your SNS topic, and sending notifications on behalf of CloudTrail to topic recipients, manually edit your CloudTrail SNS topic policy to add an `aws:SourceArn` condition key to the policy statement attached by CloudTrail. The value of this key is the ARN of the trail, or an array of trail ARNs that are using the SNS topic. Because it includes both the specific trail ID and the ID of the account that owns the trail, it restricts SNS topic access to only those accounts that have permission to manage the trail. Before you add condition keys to your SNS topic policy, get the SNS topic name from your trail's settings in the CloudTrail console.

The `aws:SourceAccount` condition key is also supported, but is not recommended.

To add the `aws:SourceArn` condition key to your SNS topic policy

1. Open the Amazon SNS console at https://console.amazonaws.cn/sns/v3/home.
2. In the navigation pane, choose **Topics**.
3. Choose the SNS topic that is shown in your trail settings, and then choose **Edit**.
4. Expand **Access policy**.
5. In the **Access policy** JSON editor, look for a block that resembles the following example.

```
    {
      "Sid": "AWSCloudTrailSNSPolicy20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-west-2:111122223333:aws-cloudtrail-
logs-111122223333-61bbe496"
    }
```

6. Add a new block for a condition, `aws:SourceArn`, as shown in the following example. The value of `aws:SourceArn` is the ARN of the trail about which you are sending notifications to SNS.

```
    {
      "Sid": "AWSCloudTrailSNSPolicy20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-west-2:111122223333:aws-cloudtrail-
logs-111122223333-61bbe496",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail3"
        }
```

```
      }
    }
```

7. When you are finished editing the SNS topic policy, choose **Save changes**.

To add the `aws:SourceAccount` condition key to your SNS topic policy

1. Open the Amazon SNS console at https://console.amazonaws.cn/sns/v3/home.
2. In the navigation pane, choose **Topics**.
3. Choose the SNS topic that is shown in your trail settings, and then choose **Edit**.
4. Expand **Access policy**.
5. In the **Access policy** JSON editor, look for a block that resembles the following example.

```
    {
      "Sid": "AWSCloudTrailSNSPolicy20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-west-2:111122223333:aws-cloudtrail-
logs-111122223333-61bbe496"
    }
```

6. Add a new block for a condition, `aws:SourceAccount`, as shown in the following example. The value of `aws:SourceAccount` is the ID of the account that owns the CloudTrail trail. This example restricts access to the SNS topic to only those users who can sign in to the Amazon account 123456789012.

```
    {
      "Sid": "AWSCloudTrailSNSPolicy20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-west-2:111122223333:aws-cloudtrail-
logs-111122223333-61bbe496",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
```

7. When you are finished editing the SNS topic policy, choose **Save changes**.

Specifying an existing topic for sending notifications

You can manually add the permissions for an Amazon SNS topic to your topic policy in the Amazon SNS console and then specify the topic in the CloudTrail console.

To manually update an SNS topic policy

1. Open the Amazon SNS console at https://console.amazonaws.cn/sns/v3/home.
2. Choose **Topics** and then choose the topic.
3. Choose **Other topic actions** and then choose **Edit topic policy**.

4. Choose **Advanced view**, and add the statement from SNS topic policy (p. 370) with the appropriate values for the Region, account ID, and topic name.
5. Choose **Update policy**.
6. If your topic is an encrypted topic, you must allow CloudTrail to have `kms:GenerateDataKey*` and the `kms:Decrypt` permissions. For more information, see Encrypted SNS topic KMS key policy (p. 370).
7. Return to the CloudTrail console and specify the topic for the trail.

Troubleshooting the SNS topic policy

The following sections describe how to troubleshoot the SNS topic policy.

Common SNS policy configuration errors

When you create a new topic as part of creating or updating a trail, CloudTrail attaches the required permissions to your topic. The topic policy uses the service principal name, `"cloudtrail.amazonaws.com"`, which allows CloudTrail to send notifications for all Regions.

If CloudTrail is not sending notifications for a Region, it's possible that your topic has an older policy that specifies CloudTrail account IDs for each Region. This policy gives CloudTrail permission to send notifications only for the Regions specified.

The following topic policy allows CloudTrail to send notifications for the specified nine Regions only:

Example topic policy with account IDs

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AWSCloudTrailSNSPolicy20131101",
        "Effect": "Allow",
        "Principal": {"AWS": [
            "arn:aws:iam::903692715234:root",
            "arn:aws:iam::035351147821:root",
            "arn:aws:iam::859597730677:root",
            "arn:aws:iam::814480443879:root",
            "arn:aws:iam::216624486486:root",
            "arn:aws:iam::086441151436:root",
            "arn:aws:iam::388731089494:root",
            "arn:aws:iam::284668455005:root",
            "arn:aws:iam::113285607260:root"
        ]},
        "Action": "SNS:Publish",
        "Resource": "aws:arn:sns:us-east-1:123456789012:myTopic"
    }]
}
```

This policy uses a permission based on individual CloudTrail account IDs. To deliver logs for a new Region, you must manually update the policy to include the CloudTrail account ID for that Region. For example, because CloudTrail added support for the US East (Ohio) Region, you must update the policy to add the account ID ARN for that Region: `"arn:aws:iam::475085895292:root"`.

As a best practice, update the policy to use a permission with the CloudTrail service principal. To do this, replace the account ID ARNs with the service principal name: `"cloudtrail.amazonaws.com"`.

This gives CloudTrail permission to send notifications for current and new Regions. The following is an updated version of the previous policy:

Example topic policy with service principal name

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AWSCloudTrailSNSPolicy20131101",
        "Effect": "Allow",
        "Principal": {"Service": "cloudtrail.amazonaws.com"},
        "Action": "SNS:Publish",
        "Resource": "arn:aws-cn:sns:us-west-2:123456789012:myTopic"
    }]
}
```

Verify that the policy has the correct values:

- In the `Resource` field, specify the account number of the topic owner. For topics that you create, specify your account number.
- Specify the appropriate values for the Region and SNS topic name.

Additional resources

For more information about SNS topics and subscribing to them, see the Amazon Simple Notification Service Developer Guide.

Troubleshooting Amazon CloudTrail identity and access

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

Topics

- I am not authorized to perform an action in CloudTrail (p. 374)
- I am not authorized to perform iam:PassRole (p. 375)
- I want to allow people outside of my Amazon Web Services account to access my CloudTrail resources (p. 375)
- I am not authorized to perform iam:PassRole (p. 375)
- I am getting a NoManagementAccountSLRExistsException exception when I try to create an organization trail or event data store (p. 376)

I am not authorized to perform an action in CloudTrail

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

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

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

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

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

If the Amazon Web Services 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 sign-in credentials.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a trail but doesn't have either the appropriate CloudTrail managed policy (**AWSCloudTrail_FullAccess** or **AWSCloudTrail_ReadOnlyAccess**) or the equivalent permissions applied to his account.

```
User: arn:aws-cn:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudtrail:GetTrailStatus on resource: My-Trail
```

In this case, Mateo asks his administrator to update his policies to allow him to access trail information and status in the console.

If you sign in with an IAM user or role that has the **AWSCloudTrail_FullAccess** managed policy or its equivalent permissions, and you can't configure Amazon Config or Amazon CloudWatch Logs integration with a trail, you might be missing the required permissions for integration with those services. For more information, see Granting permission to view Amazon Config information on the CloudTrail console (p. 360) and Granting permission to view and configure Amazon CloudWatch Logs information on the CloudTrail console (p. 360).

I am not authorized to perform `iam:PassRole`

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

Some Amazon Web 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 CloudTrail. 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-cn: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 Amazon administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my Amazon Web Services account to access my CloudTrail resources

You can create a role and share CloudTrail information between multiple Amazon Web Services accounts. For more information, see Sharing CloudTrail log files between Amazon accounts (p. 297).

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 CloudTrail supports these features, see How Amazon CloudTrail works with IAM (p. 344).
- To learn how to provide access to your resources across Amazon Web Services accounts that you own, see Providing access to an IAM user in another Amazon Web Services account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party Amazon Web Services accounts, see Providing access to Amazon Web Services 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 How IAM roles differ from resource-based policies in the *IAM User Guide*.

I am not authorized to perform `iam:PassRole`

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

Some Amazon Web 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 CloudTrail. 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-cn: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 Amazon administrator. Your administrator is the person who provided you with your sign-in credentials.

I am getting a `NoManagementAccountSLRExistsException` exception when I try to create an organization trail or event data store

The `NoManagementAccountSLRExistsException` exception is thrown when the management account does not have a service-linked role. When you add a delegated administrator using the Amazon Organizations Amazon CLI or API operation, the service-linked role doesn't get created if it does not exist.

When you use your organization's management account to add a delegated administrator or create an organization trail or event data store in the CloudTrail console, or by using the Amazon CLI or CloudTrail API, CloudTrail automatically creates a service-linked role for your management account if one does not already exist.

If you haven't added a delegated administrator, use the CloudTrail console, Amazon CLI or CloudTrail API to add the delegated administrator. For more information about adding a delegated administrator, see Add a CloudTrail delegated administrator (p. 337) and RegisterOrganizationDelegatedAdmin (API).

If you've already added the delegated administrator, use the management account to create the organization trail or event data store in the CloudTrail console, or by using the Amazon CLI or CloudTrail API. For more information about creating an organization trail, see Creating a trail for your organization in the console (p. 130), Creating a trail for an organization with the Amazon Command Line Interface (p. 135), and CreateTrail (API).

Using service-linked roles for Amazon CloudTrail

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

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

For information about other services that support service-linked roles, see Amazon Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for CloudTrail

CloudTrail uses the service-linked role named **AWSServiceRoleForCloudTrail** – This service linked role is used for supporting the organization trail feature.

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

- `cloudtrail.amazonaws.com`

This role is used to support the creation and management of organization trails in CloudTrail. For more information, see Creating a trail for an organization (p. 125).

The role permissions policy allows CloudTrail to complete the following actions on the specified resources:

- Actions on all CloudTrail resources:
  - `All`
- Actions on all Organizations resources:
  - `organizations:DescribeAccount`
  - `organizations:DescribeOrganization`
  - `organizations:ListAccounts`
  - `organizations:ListAWSServiceAccessForOrganization`
- Actions on all Organizations resources for the CloudTrail service principal:
  - `organizations:ListDelegatedAdministrators`

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 a service-linked role for CloudTrail

You don't need to manually create a service-linked role. When you create an organization trail or add a delegated administrator in the CloudTrail console, or by using the Amazon CLI or API operation, CloudTrail creates the service-linked role for you if it does not already exist.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create an organization trail or add a delegated administrator, CloudTrail creates the service-linked role for you again.

Editing a service-linked role for CloudTrail

CloudTrail does not allow you to edit the AWSServiceRoleForCloudTrail service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for CloudTrail

You don't need to manually delete the AWSServiceRoleForCloudTrail role. If an Amazon account is removed from an Organizations organization, the AWSServiceRoleForCloudTrail role is automatically removed from that Amazon account. You cannot detach or remove policies from the AWSServiceRoleForCloudTrail service-linked role in an organization management account without removing the account from the organization.

You can also use the IAM console, the Amazon CLI or the Amazon API to manually delete the service-linked role. To do this, you must first manually clean up the resources for your service-linked role, and then you can manually delete it.

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

To remove a resource being used by the AWSServiceRoleForCloudTrail role, you can do one of the following:

- Remove the Amazon account from the organization in Organizations.
- Update the trail so that it is no longer an organization trail.
- Delete the trail.

For more information, see Creating a trail for an organization (p. 125), Updating a trail (p. 89), and Deleting a trail (p. 96).

To manually delete the service-linked role using IAM

Use the IAM console, the Amazon CLI, or the Amazon API to delete the AWSServiceRoleForCloudTrail service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for CloudTrail service-linked roles

CloudTrail supports using service-linked roles in all of the Regions where CloudTrail and Organizations are both available. For more information, see Amazon Regions and Endpoints.

Amazon managed policies for Amazon CloudTrail

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

Amazon services maintain and update Amazon managed policies. You can't change the permissions in Amazon managed policies. Services occasionally add additional permissions to an Amazon managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an Amazon managed policy when a new feature

is launched or when new operations become available. Services do not remove permissions from an Amazon managed policy, so policy updates won't break your existing permissions.

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

Amazon managed policy: AWSCloudTrail_ReadOnlyAccess

A user identity that has the `AWSCloudTrail_ReadOnlyAccess` policy attached to its role can perform read-only actions in CloudTrail, such as `Get*`, `List*`, and `Describe*` actions on trails, CloudTrail Lake event data stores, or Lake queries.

Permissions details

This policy includes the following permissions.

- `cloudtrail` – Allows user identities to perform read-only API operations in the CloudTrail console, the Amazon SDKs, and the Amazon CLI, including `Get*` operations such as `GetEventDataStores`, `List*` operations such as `ListTrails`, and `Describe*` operations such as `DescribeTrails`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:Get*",
                "cloudtrail:Describe*",
                "cloudtrail:List*",
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudTrail updates to Amazon managed policies

View details about updates to Amazon managed policies for CloudTrail. For automatic alerts about changes to this page, subscribe to the RSS feed on the CloudTrail Document history (p. 438) page.

Change	Description	Date
AWSCloudTrail_ReadOnlyAccess (p. 379) – Update to an existing policy	CloudTrail changed the name of the `AWSCloudTrailReadOnlyAccess` policy to `AWSCloudTrail_ReadOnlyAccess`. Also, the scope of permissions in the policy has been reduced to CloudTrail actions. It no longer includes Amazon S3, Amazon KMS, or Amazon Lambda action permissions.	June 6, 2022

Change	Description	Date
CloudTrail started tracking changes	CloudTrail started tracking changes for its Amazon managed policies.	June 6, 2022

Compliance validation for Amazon CloudTrail

Third-party auditors assess the security and compliance of Amazon CloudTrail as part of multiple Amazon compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

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

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

Your compliance responsibility when using Amazon Web Services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. Amazon provides the following resources to help with compliance:

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

Resilience in Amazon CloudTrail

The Amazon global infrastructure is built around Amazon Regions and Availability Zones. Amazon Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures. If you specifically need to replicate your CloudTrail log files over greater geographic distances, you can use Cross-Region Replication for your trail Amazon S3 buckets, which enables automatic, asynchronous copying of objects across buckets in different Amazon Regions.

For more information about Amazon Regions and Availability Zones, see Amazon Global Infrastructure.

In addition to the Amazon global infrastructure, CloudTrail offers several features to help support your data resiliency and backup needs.

Trails and event data stores that log events in all Amazon Regions

When you apply a trail to all Amazon Regions, CloudTrail creates trails with identical configurations in all other Amazon Regions in your account. When Amazon adds a new Region, that trail configuration is automatically created in the new Region.

When you create a multi-Region event data store, CloudTrail collects events that occur in all Amazon Web Services Regions.

Versioning, lifecycle configuration, and object lock protection for CloudTrail log data

Because CloudTrail uses Amazon S3 buckets to store log files, you can also use the features provided by Amazon S3 to help support your data resiliency and backup needs. For more information, see Resilience in Amazon S3.

Infrastructure security in Amazon CloudTrail

As a managed service, Amazon CloudTrail is protected by Amazon global network security. For information about Amazon security services and how Amazon protects infrastructure, see Amazon Cloud Security. To design your Amazon environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar Amazon Well-Architected Framework*.

You use Amazon published API calls to access CloudTrail 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 Amazon Security Token Service (Amazon STS) to generate temporary security credentials to sign requests.

The following security best practices also address infrastructure security in CloudTrail:

- Consider Amazon VPC endpoints for trail access. (p. 152)
- Consider Amazon VPC endpoints for Amazon S3 bucket access. For more information, see Example Bucket Policies for VPC Endpoints for Amazon S3.
- Identify and audit all Amazon S3 buckets that contain CloudTrail log files. Consider using tags to help identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources. For more information, see Amazon Resource Groups.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In Amazon, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, Amazon provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Amazon CloudTrail gives another service to the resource.

Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (*) for the unknown portions of the ARN. For example, `"arn:aws-cn:cloudtrail:*:AccountID:trail/*"`. When you include a wildcard, you must also use the `StringLike` condition operator.

The value of `aws:SourceArn` must be the ARN of the trail, event data store, or channel that is using the resource.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in CloudTrail to prevent the confused deputy problem: Amazon S3 bucket policy for CloudTrail Lake query results (p. 368).

Security best practices in Amazon CloudTrail

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

Topics

- CloudTrail detective security best practices (p. 382)
- CloudTrail preventative security best practices (p. 383)

CloudTrail detective security best practices

Create a trail

For an ongoing record of events in your Amazon account, you must create a trail. Although CloudTrail provides 90 days of event history information for management events in the CloudTrail console without creating a trail, it is not a permanent record, and it does not provide information about all possible types of events. For an ongoing record, and for a record that contains all the event types you specify, you must create a trail, which delivers log files to an Amazon S3 bucket that you specify.

To help manage your CloudTrail data, consider creating one trail that logs management events in all Amazon Web Services Regions, and then creating additional trails that log specific event types for resources, such as Amazon S3 bucket activity or Amazon Lambda functions.

The following are some steps you can take:

- Create a trail for your Amazon account. (p. 78)
- Create a trail for an organization. (p. 125)

Apply trails to all Amazon Web Services Regions

To obtain a complete record of events taken by an IAM identity, or service in your Amazon account, each trail should be configured to log events in all Amazon Web Services Regions. By logging events

in all Amazon Web Services Regions, you ensure that all events that occur in your Amazon account are logged, regardless of which Amazon Region where they occurred. This includes logging global service events (p. 14), which are logged to an Amazon Region specific to that service. When you create a trail that applies to all Regions, CloudTrail records events in each Region and delivers the CloudTrail event log files to an S3 bucket that you specify. If an Amazon Region is added after you create a trail that applies to all Regions, that new Region is automatically included, and events in that Region are logged. This is the default option when you create a trail in the CloudTrail console.

The following are some steps you can take:

- Create a trail for your Amazon account. (p. 78)
- Update an existing trail (p. 89)to log events in all Amazon Web Services Regions.
- Implement ongoing detective controls to help ensure all trails created are logging events in all Amazon Web Services Regions by using the multi-region-cloud-trail-enabled rule in Amazon Config.

Enable CloudTrail log file integrity

Validated log files are especially valuable in security and forensic investigations. For example, a validated log file enables you to assert positively that the log file itself has not changed, or that particular IAM identity credentials performed specific API activity. The CloudTrail log file integrity validation process also lets you know if a log file has been deleted or changed, or assert positively that no log files were delivered to your account during a given period of time. CloudTrail log file integrity validation uses industry standard algorithms: SHA-256 for hashing and SHA-256 with RSA for digital signing. This makes it computationally unfeasible to modify, delete or forge CloudTrail log files without detection. For more information, see Enabling validation and validating files (p. 308).

Integrate with Amazon CloudWatch Logs

CloudWatch Logs allows you to monitor and receive alerts for specific events captured by CloudTrail. The events sent to CloudWatch Logs are those configured to be logged by your trail, so make sure you have configured your trail or trails to log the event types (management events and/or data events) that you are interested in monitoring.

For example, you can monitor key security and network-related management events, such as failed Amazon Web Services Management Console sign-in events (p. 284).

The following are some steps you can take:

- Review example CloudWatch Logs integrations for CloudTrail (p. 282).
- Configure your trail to send events to CloudWatch Logs (p. 277).
- Consider implementing ongoing detective controls to help ensure all trails are sending events to CloudWatch Logs for monitoring by using the cloud-trail-cloud-watch-logs-enabled rule in Amazon Config.

Use Amazon Security Hub

Monitor your usage of CloudTrail as it relates to security best practices by using Amazon Security Hub. Security Hub uses detective *security controls* to evaluate resource configurations and *security standards* to help you comply with various compliance frameworks. For more information about using Security Hub to evaluate CloudTrail resources, see Amazon CloudTrail controls in the *Amazon Security Hub User Guide*.

CloudTrail preventative security best practices

The following best practices for CloudTrail can help prevent security incidents.

Log to a dedicated and centralized Amazon S3 bucket

CloudTrail log files are an audit log of actions taken by an IAM identity or an Amazon service. The integrity, completeness and availability of these logs is crucial for forensic and auditing purposes. By logging to a dedicated and centralized Amazon S3 bucket, you can enforce strict security controls, access, and segregation of duties.

The following are some steps you can take:

- Create a separate Amazon account as a log archive account. If you use Amazon Organizations, enroll this account in the organization, and consider creating an organization trail (p. 125) to log data for all Amazon accounts in your organization.
- If you do not use Organizations but want to log data for multiple Amazon accounts, create a trail (p. 78) to log activity in this log archive account. Restrict access to this account to only trusted administrative users who should have access to account and auditing data.
- As part of creating a trail, whether it is an organization trail or a trail for a single Amazon account, create a dedicated Amazon S3 bucket to store log files for this trail.
- If you want to log activity for more than one Amazon account, modify the bucket policy (p. 291) to allow logging and storing log files for all Amazon accounts that you want to log Amazon account activity.
- If you are not using an organization trail, create trails in all of your Amazon accounts, specifying the Amazon S3 bucket in the log archive account.

Use server-side encryption with Amazon KMS managed keys

By default, the log files delivered by CloudTrail to your bucket are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). To provide a security layer that is directly manageable, you can instead use server-side encryption with Amazon KMS–managed keys (SSE-KMS) for your CloudTrail log files. To use SSE-KMS with CloudTrail, you create and manage an Amazon KMS key, also known as a KMS key.

> **Note**
> If you use SSE-KMS and log file validation, and you have modified your Amazon S3 bucket policy to only allow SSE-KMS encrypted files, you will not be able to create trails that utilize that bucket unless you modify your bucket policy to specifically allow AES256 encryption, as shown in the following example policy line.

```
"StringNotEquals": { "s3:x-amz-server-side-encryption": ["aws:kms", "AES256"] }
```

The following are some steps you can take:

- Review the advantages of encrypting your log files with SSE-KMS (p. 386).
- Create a KMS key to use for encrypting log files (p. 387).
- Configure log file encryption for your trails. (p. 394)
- Consider implementing ongoing detective controls to help ensure all trails are encrypting log files with SSE-KMS by using the cloud-trail-encryption-enabled rule in Amazon Config.

Add a condition key to the default Amazon SNS topic policy

When you configure a trail to send notifications to Amazon SNS, CloudTrail adds a policy statement to your SNS topic access policy that allows CloudTrail to send content to an SNS topic. As a security best practice, we recommend adding an `aws:SourceArn` (or optionally `aws:SourceAccount`) condition key to the CloudTrail policy statement. This helps prevent unauthorized account access to your SNS topic. For more information, see Amazon SNS topic policy for CloudTrail (p. 369).

Implement least privilege access to Amazon S3 buckets where you store log files

CloudTrail trails log events to an Amazon S3 bucket that you specify. These log files contain an audit log of actions taken by IAM identities and Amazon services. The integrity and completeness of these log files are crucial for auditing and forensic purposes. In order to help ensure that integrity, you should adhere to the principle of least privilege when creating or modifying access to any Amazon S3 bucket used for storing CloudTrail log files.

Take the following steps:

- Review the Amazon S3 bucket policy (p. 363) for any and all buckets where you store log files and adjust it if necessary to remove any unnecessary access. This bucket policy will be generated for you if you create a trail using the CloudTrail console, but can also be created and managed manually.
- As a security best practice, be sure to manually add a `aws:SourceArn` condition key to the bucket policy. For more information, see Amazon S3 bucket policy for CloudTrail (p. 363).
- If you are using the same Amazon S3 bucket to store log files for multiple Amazon accounts, follow the guidance for receiving log files for multiple accounts (p. 290).
- If you are using an organization trail, make sure you follow the guidance for organization trails (p. 125), and review the example policy for an Amazon S3 bucket for an organization trail in Creating a trail for an organization with the Amazon Command Line Interface (p. 135).
- Review the Amazon S3 security documentation and the example walkthrough for securing a bucket.

Enable MFA delete on the Amazon S3 bucket where you store log files

When you configure multi-factor authentication (MFA), attempts to change the versioning state of bucket, or delete an object version in a bucket, require additional authentication. This way, even if a user acquires the password of an IAM user with permissions to permanently delete Amazon S3 objects, you can still prevent operations that could compromise your log files.

The following are some steps you can take:

- Review the MFA delete guidance in the *Amazon Simple Storage Service User Guide*.
- Add an Amazon S3 bucket policy to require MFA.

> **Note**
> You cannot use MFA delete with lifecycle configurations. For more information about lifecycle configurations and how they interact with other configurations, see Lifecycle and other bucket configurations in the *Amazon Simple Storage Service User Guide*.

Configure object lifecycle management on the Amazon S3 bucket where you store log files

The CloudTrail trail default is to store log files indefinitely in the Amazon S3 bucket configured for the trail. You can use the Amazon S3 object lifecycle management rules to define your own retention policy to better meet your business and auditing needs. For example, you might want to archive log files that are more than a year old to Amazon Glacier, or delete log files after a certain amount of time has passed.

> **Note**
> Lifecycle configuration on multi-factor authentication (MFA)-enabled buckets is not supported.

Limit access to the AWSCloudTrail_FullAccess policy

Users with the AWSCloudTrail_FullAccess (p. 358) policy have the ability to disable or reconfigure the most sensitive and important auditing functions in their Amazon accounts. This policy is not intended to be shared or applied broadly to IAM identities in your Amazon account. Limit application of this policy to as few individuals as possible, those you expect to act as Amazon account administrators.

Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS)

By default, the log files delivered by CloudTrail to your bucket are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). To provide a security layer that is directly manageable, you can instead use server-side encryption with Amazon KMS keys (SSE-KMS) for your CloudTrail log files.

> **Note**
> Enabling server-side encryption encrypts the log files but not the digest files with SSE-KMS. Digest files are encrypted with Amazon S3-managed encryption keys (SSE-S3).
> If you are using an existing S3 bucket with an S3 Bucket Key, CloudTrail must be allowed permission in the key policy to use the Amazon KMS actions `GenerateDataKey` and `DescribeKey`. If `cloudtrail.amazonaws.com` is not granted those permissions in the key policy, you cannot create or update a trail.

To use SSE-KMS with CloudTrail, you create and manage a KMS key, also known as an Amazon KMS key. You attach a policy to the key that determines which users can use the key for encrypting and decrypting CloudTrail log files. The decryption is seamless through S3. When authorized users of the key read CloudTrail log files, S3 manages the decryption, and the authorized users are able to read log files in unencrypted form.

This approach has the following advantages:

- You can create and manage the KMS key encryption keys yourself.
- You can use a single KMS key to encrypt and decrypt log files for multiple accounts across all Regions.
- You have control over who can use your key for encrypting and decrypting CloudTrail log files. You can assign permissions for the key to the users in your organization according to your requirements.
- You have enhanced security. With this feature, to read log files, the following permissions are required:
  - A user must have S3 read permissions for the bucket that contains the log files.
  - A user must also have a policy or role applied that allows decrypt permissions by the KMS key policy.
- Because S3 automatically decrypts the log files for requests from users authorized to use the KMS key, SSE-KMS encryption for CloudTrail log files is backward-compatible with applications that read CloudTrail log data.

> **Note**
> The KMS key that you choose must be created in the same Amazon Region as the Amazon S3 bucket that receives your log files. For example, if the log files will be stored in a bucket in the US East (Ohio) Region, you must create or choose a KMS key that was created in that Region. To verify the Region for an Amazon S3 bucket, inspect its properties in the Amazon S3 console.

Enabling log file encryption

> **Note**
> If you create a KMS key in the CloudTrail console, CloudTrail adds the required KMS key policy sections for you. Follow these procedures if you created a key in the IAM console or Amazon CLI and you need to manually add the required policy sections.

To enable SSE-KMS encryption for CloudTrail log files, perform the following high-level steps:

1. Create a KMS key.
   - For information about creating a KMS key with the Amazon Web Services Management Console, see Creating Keys in the *Amazon Key Management Service Developer Guide*.

- For information about creating a KMS key with the Amazon CLI, see create-key.

  Note
  The KMS key that you choose must be in the same Region as the S3 bucket that receives your log files. To verify the Region for an S3 bucket, inspect the bucket's properties in the S3 console.

2. Add policy sections to the key that enable CloudTrail to encrypt and users to decrypt log files.

   - For information about what to include in the policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).

     Warning
     Be sure to include decrypt permissions in the policy for all users who need to read log files. If you do not perform this step before adding the key to your trail configuration, users without decrypt permissions cannot read encrypted files until you grant them those permissions.

   - For information about editing a policy with the IAM console, see Editing a Key Policy in the *Amazon Key Management Service Developer Guide*.
   - For information about attaching a policy to a KMS key with the Amazon CLI, see put-key-policy.

3. Update your trail to use the KMS key whose policy you modified for CloudTrail.

   - To update your trail configuration by using the CloudTrail console, see Updating a resource to use your KMS key (p. 394).
   - To update your trail configuration by using the Amazon CLI, see Enabling and disabling CloudTrail log file encryption with the Amazon CLI (p. 396).

CloudTrail also supports Amazon KMS multi-Region keys. For more information about multi-Region keys, see Using multi-Region keys in the *Amazon Key Management Service Developer Guide*.

The next section describes the policy sections that your KMS key policy requires for use with CloudTrail.

Granting permissions to create a KMS key

You can grant users permission to create an Amazon KMS key with the `AWSKeyManagementServicePowerUser` policy.

To grant permission to create a KMS key

1. Open the IAM console at https://console.amazonaws.cn/iam/.
2. Choose the group or user that you want to give permission.
3. Choose **Permissions**, and then choose **Attach Policy**.
4. Search for **AWSKeyManagementServicePowerUser**, choose the policy, and then choose **Attach policy**.

   The user now has permission to create a KMS key. If you want to create custom policies for your users, see Creating Customer Managed Policies in the *IAM User Guide*.

Configure Amazon KMS key policies for CloudTrail

You can create an Amazon KMS key in three ways:

- The CloudTrail console

- The Amazon Management console
- The Amazon CLI

> **Note**
> If you create a KMS key in the CloudTrail console, CloudTrail adds the required KMS key policy for you. You do not need to manually add the policy statements. See Default KMS key policy created in CloudTrail console (p. 394).

If you create a KMS key in the Amazon Management or the Amazon CLI, you must add policy sections to the key so that you can use it with CloudTrail. The policy must allow CloudTrail to use the key to encrypt your log files and event data stores, and allow the users you specify to read log files in unencrypted form.

See the following resources:

- To create a KMS key with the Amazon CLI, see create-key.
- To edit a KMS key policy for CloudTrail, see Editing a Key Policy in the *Amazon Key Management Service Developer Guide*.
- For technical details on how CloudTrail uses Amazon KMS, see How Amazon CloudTrail Uses Amazon KMS in the *Amazon Key Management Service Developer Guide*.

Required KMS key policy sections for use with CloudTrail

If you created a KMS key with the Amazon Management console or the Amazon CLI, then you must, at minimum, add the following statements to your KMS key policy for it to work with CloudTrail.

Topics

- Required KMS key policy elements for trails (p. 388)
- Required KMS key policy elements for event data stores (p. 389)

Required KMS key policy elements for trails

1. Enable CloudTrail log encrypt permissions. See Granting encrypt permissions (p. 389).
2. Enable CloudTrail log decrypt permissions. See Granting decrypt permissions (p. 390). If you are using an existing S3 bucket with an S3 Bucket Key, `kms:Decrypt` permissions are required to create or update a trail with SSE-KMS encryption enabled.
3. Enable CloudTrail to describe KMS key properties. See Enable CloudTrail to describe KMS key properties (p. 393).

As a security best practice, add an `aws:SourceArn` condition key to the KMS key policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail uses the KMS key only for a specific trail or trails. The value of `aws:SourceArn` is always the trail ARN (or array of trail ARNs) that is using the KMS key. Be sure to add the `aws:SourceArn` condition key to KMS key policies for existing trails.

The `aws:SourceAccount` condition key is also supported, but not recommended. The value of `aws:SourceAccount` is the account ID of the trail owner, or for organization trails, the management account ID.

> **Important**
> When you add the new sections to your KMS key policy, do not change any existing sections in the policy.
> If encryption is enabled on a trail, and the KMS key is disabled, or the KMS key policy is not correctly configured for CloudTrail, CloudTrail cannot deliver logs.

Required KMS key policy elements for event data stores

1. Enable CloudTrail log encrypt permissions. See Granting encrypt permissions (p. 389).
2. Enable CloudTrail log decrypt permissions. See Granting decrypt permissions (p. 390).
3. Grant users and roles permission to encrypt and decrypt event data store data with the KMS key.

   When you create an event data store and encrypt it with a KMS key, or run queries on an event data store that you're encrypting with a KMS key, you should have write access to the KMS key. The KMS key policy must have access to CloudTrail, and the KMS key should be manageable by users who run operations (such as queries) on the event data store.
4. Enable CloudTrail to describe KMS key properties. See Enable CloudTrail to describe KMS key properties (p. 393).

The `aws:SourceArn` and `aws:SourceAccount` condition keys are not supported in KMS key policies for event data stores.

Important
When you add the new sections to your KMS key policy, do not change any existing sections in the policy.
If encryption is enabled on an event data store, and the KMS key is disabled or deleted, or the KMS key policy is not correctly configured for CloudTrail, CloudTrail cannot deliver events to your event data store.

Granting encrypt permissions

Example Allow CloudTrail to encrypt logs on behalf of specific accounts

CloudTrail needs explicit permission to use the KMS key to encrypt logs on behalf of specific accounts. To specify an account, add the following required statement to your KMS key policy and replace *account-id*, *region*, and *trailName* with the appropriate values for your configuration. You can add additional account IDs to the `EncryptionContext` section to enable those accounts to use CloudTrail to use your KMS key to encrypt log files.

As a security best practice, add an `aws:SourceArn` condition key to the KMS key policy for a trail. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail uses the KMS key only for a specific trail or trails.

```
{
  "Sid": "Allow CloudTrail to encrypt logs",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudtrail.amazonaws.com"
  },
  "Action": "kms:GenerateDataKey*",
  "Resource": "arn:aws-cn:kms:region:account-id:key/key-id",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:cloudtrail:arn": [
        "arn:aws-cn:cloudtrail:*:account-id:trail/*"
      ]
    },
    "StringEquals": {
        "aws:SourceArn": "arn:aws:cloudtrail:region:account-id:trail/trail-name"
    }
  }
}
```

Example

The following example policy statement illustrates how another account can use your KMS key to encrypt CloudTrail logs.

Scenario

- Your KMS key is in account *111111111111*.
- Both you and account *222222222222* will encrypt logs.

In the policy, you add one or more accounts that encrypt with your key to the CloudTrail **EncryptionContext**. This restricts CloudTrail to using your key to encrypt logs only for the accounts that you specify. When you give the root of account *222222222222* permission to encrypt logs, it delegates permission to the account administrator to encrypt the necessary permissions to other users in that account. The account administrator does this by changing the policies associated with those IAM users.

As a security best practice, add an `aws:SourceArn` condition key to the KMS key policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail uses the KMS key only for the specified trails. This condition isn't supported in KMS key policies for event data stores.

KMS key policy statement:

```
{
  "Sid": "Enable CloudTrail encrypt permissions",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudtrail.amazonaws.com"
  },
  "Action": "kms:GenerateDataKey*",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:cloudtrail:arn": [
        "arn:aws-cn:cloudtrail:*:111111111111:trail/*",
        "arn:aws-cn:cloudtrail:*:222222222222:trail/*"
      ]
    },
    "StringEquals": {
        "aws:SourceArn": "arn:aws:cloudtrail:region:account-id:trail/trail-name"
    }
  }
}
```

For more information about editing a KMS key policy for use with CloudTrail, see Editing a key policy in the Amazon Key Management Service Developer Guide.

Granting decrypt permissions

Before you add your KMS key to your CloudTrail configuration, it is important to give decrypt permissions to all users who require them. Users who have encrypt permissions but no decrypt permissions cannot read encrypted logs. If you are using an existing S3 bucket with an S3 Bucket Key, `kms:Decrypt` permissions are required to create or update a trail with SSE-KMS encryption enabled.

Enable CloudTrail log decrypt permissions

Users of your key must be given explicit permissions to read the log files that CloudTrail has encrypted. To enable users to read encrypted logs, add the following required statement to your KMS key policy, modifying the `Principal` section to add a line for every principal that you want to be able decrypt by using your KMS key.

```
{
  "Sid": "Enable CloudTrail log decrypt permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws-cn:iam::account-id:user/username"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:cloudtrail:arn": "false"
    }
  }
}
```

The following is an example policy that is required to allow the CloudTrail service principal to decrypt trail logs.

```
{
      "Sid": "Allow CloudTrail to decrypt a trail",
      "Effect": "Allow",
      "Principal": {
          "Service": "cloudtrail.amazonaws.com"
        },
      "Action": "kms:Decrypt",
      "Resource": "*"
}
```

Allow users in your account to decrypt trail logs with your KMS key

Example

This policy statement illustrates how to allow a user or role in your account to use your key to read the encrypted logs in your account's S3 bucket.

Example Scenario

- Your KMS key, S3 bucket, and IAM user Bob are in account *111111111111*.
- You give IAM user Bob permission to decrypt CloudTrail logs in the S3 bucket.

In the key policy, you enable CloudTrail log decrypt permissions for IAM user Bob.

KMS key policy statement:

```
{
  "Sid": "Enable CloudTrail log decrypt permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws-cn:iam::111111111111:user/Bob"
  },
  "Action": "kms:Decrypt",
  "Resource": "arn:aws-cn:kms:region:account-id:key/key-id",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:cloudtrail:arn": "false"
    }
  }
}
```

Allow users in other accounts to decrypt trail logs with your KMS key

You can allow users in other accounts to use your KMS key to decrypt trail logs, but not event data store logs. The changes required to your key policy depend on whether the S3 bucket is in your account or in another account.

Allow users of a bucket in a different account to decrypt logs

Example

This policy statement illustrates how to allow an IAM user or role in another account to use your key to read encrypted logs from an S3 bucket in the other account.

Scenario

- Your KMS key is in account *111111111111*.
- The IAM user Alice and S3 bucket are in account *222222222222*.

In this case, you give CloudTrail permission to decrypt logs under account *222222222222*, and you give Alice's IAM user policy permission to use your key *KeyA*, which is in account *111111111111*.

KMS key policy statement:

```
{
  "Sid": "Enable encrypted CloudTrail log read access",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws-cn:iam::222222222222:root"
    ]
  },
  "Action": "kms:Decrypt",
  "Resource": "arn:aws-cn:kms:region:account-id:key/key-id",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:cloudtrail:arn": "false"
    }
  }
}
```

Alice's IAM user policy statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws-cn:kms:us-west-2:111111111111:key/KeyA"
    }
  ]
}
```

Allow users in a different account to decrypt trail logs from your bucket

Example

This policy illustrates how another account can use your key to read encrypted logs from your S3 bucket.

Example Scenario

- Your KMS key and S3 bucket are in account *111111111111*.
- The user who reads logs from your bucket is in account *222222222222*.

To enable this scenario, you enable decrypt permissions for the IAM role **CloudTrailReadRole** in your account, and then give the other account permission to assume that role.

KMS key policy statement:

```
{
  "Sid": "Enable encrypted CloudTrail log read access",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws-cn:iam::11111111111:role/CloudTrailReadRole"
    ]
  },
  "Action": "kms:Decrypt",
  "Resource": "arn:aws-cn:kms:region:account-id:key/key-id",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:cloudtrail:arn": "false"
    }
  }
}
```

CloudTrailReadRole trust entity policy statement:

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Sid": "Allow CloudTrail access",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws-cn:iam::222222222222:root"
     },
     "Action": "sts:AssumeRole"
   }
 ]
}
```

For information about editing a KMS key policy for use with CloudTrail, see Editing a Key Policy in the *Amazon Key Management Service Developer Guide*.

Enable CloudTrail to describe KMS key properties

CloudTrail requires the ability to describe the properties of the KMS key. To enable this functionality, add the following required statement as is to your KMS key policy. This statement does not grant CloudTrail any permissions beyond the other permissions that you specify.

As a security best practice, add an `aws:SourceArn` condition key to the KMS key policy. The IAM global condition key `aws:SourceArn` helps ensure that CloudTrail uses the KMS key only for a specific trail or trails.

```
{
  "Sid": "Allow CloudTrail access",
  "Effect": "Allow",
  "Principal": {
```

```
    "Service": "cloudtrail.amazonaws.com"
  },
  "Action": "kms:DescribeKey",
  "Resource": "arn:aws-cn:kms:region:account-id:key/key-id",
  "Condition": {
    "StringEquals": {
        "aws:SourceArn": "arn:aws:cloudtrail:region:account-id:trail/trail-name"
    }
  }
}
```

For more information about editing KMS key policies, see Editing a Key Policy in the *Amazon Key Management Service Developer Guide*.

Default KMS key policy created in CloudTrail console

If you create an Amazon KMS key in the CloudTrail console, the following policies are automatically created for you. The policy allows these permissions:

- Allows Amazon Web Services account (root) permissions for the KMS key.
- Allows CloudTrail to encrypt log files under the KMS key and describe the KMS key.
- Allows all users in the specified accounts to decrypt log files.
- Allows all users in the specified account to create a KMS alias for the KMS key.
- Enables cross-account log decryption for the account ID of the account that created the trail.

Topics

- Default KMS key policy for trails (p. 394)

Default KMS key policy for trails

The following is the default policy created for a Amazon KMS key that you use with a trail.

> **Note**
> The policy includes a statement to allow cross accounts to decrypt log files with the KMS key.

Updating a resource to use your KMS key

In the Amazon CloudTrail console, update a trail or an event data store to use an Amazon Key Management Service key. Be aware that using your own KMS key incurs Amazon KMS costs for encryption and decryption. For more information, see Amazon Key Management Service Pricing.

Topics

- Update a trail to use a KMS key (p. 394)
- Update an event data store to use a KMS key (p. 395)

Update a trail to use a KMS key

To update a trail to use the Amazon KMS key that you modified for CloudTrail, complete the following steps in the CloudTrail console.

> **Note**
> Updating a trail with the following procedure encrypts the log files but not the digest files with SSE-KMS. Digest files are encrypted with Amazon S3-managed encryption keys (SSE-S3).
> If you are using an existing S3 bucket with an S3 Bucket Key, CloudTrail must be allowed permission in the key policy to use the Amazon KMS actions `GenerateDataKey` and

`DescribeKey`. If `cloudtrail.amazonaws.com` is not granted those permissions in the key policy, you cannot create or update a trail.

To update a trail using the Amazon CLI, see Enabling and disabling CloudTrail log file encryption with the Amazon CLI (p. 396).

To update a trail to use your KMS key

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. Choose **Trails** and then choose a trail name.
3. In **General details**, choose **Edit**.
4. For **Log file SSE-KMS encryption**, choose **Enabled** if you want to encrypt your log files using SSE-KMS encryption instead of SSE-S3 encryption. The default is **Enabled**. If you don't enable SSE-KMS encryption, your logs are encrypted using SSE-S3 encryption. For more information about SSE-KMS encryption, see Using server-side encryption with Amazon Key Management Service (SSE-KMS). For more information about SSE-S3 encryption, see Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3).

   Choose **Existing** to update your trail with your Amazon KMS key. Choose a KMS key that is in the same Region as the S3 bucket that receives your log files. To verify the Region for an S3 bucket, view its properties in the S3 console.

   Note
   You can also type the ARN of a key from another account. For more information, see Updating a resource to use your KMS key (p. 394). The key policy must allow CloudTrail to use the key to encrypt your log files, and allow the users you specify to read log files in unencrypted form. For information about manually editing the key policy, see Configure Amazon KMS key policies for CloudTrail (p. 387).

   In **Amazon KMS Alias**, specify the alias for which you changed the policy for use with CloudTrail, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394).

   You can type the alias name, ARN, or the globally unique key ID. If the KMS key belongs to another account, verify that the key policy has permissions that enable you to use it. The value can be one of the following formats:

   - **Alias Name**: `alias/MyAliasName`
   - **Alias ARN**: `arn:aws:kms:region:123456789012:alias/MyAliasName`
   - **Key ARN**: `arn:aws:kms:region:123456789012:key/12345678-1234-1234-1234-123456789012`
   - **Globally unique key ID**: `12345678-1234-1234-1234-123456789012`
5. Choose **Update trail**.

   Note
   If the KMS key that you chose is disabled or is pending deletion, you cannot save the trail with that KMS key. You can enable the KMS key or choose another one. For more information, see Key state: Effect on your KMS key in the *Amazon Key Management Service Developer Guide*.

Update an event data store to use a KMS key

To update an event data store to use the Amazon KMS key that you modified for CloudTrail, complete the following steps in the CloudTrail console.

To update an event data store by using the Amazon CLI, see Update an event data store (p. 222).

Important
Disabling or deleting the KMS key, or removing CloudTrail permissions on the key, prevents CloudTrail from ingesting events into the event data store, and prevents users from querying data in the event data store that was encrypted with the key. After you associate an event data store with a KMS key, the KMS key cannot be removed or changed. Before you disable or delete a KMS key that you are using with an event data store, delete or back up your event data store.

To update an event data store to use your KMS key

1. Sign in to the Amazon Web Services Management Console and open the CloudTrail console at https://console.amazonaws.cn/cloudtrail/.
2. In the navigation pane, choose **Event data stores** in **Lake**. Choose an event data store to update.
3. In **General details**, choose **Edit**.
4. For **Encryption**, if it is not already enabled, choose **Use my own Amazon KMS key** to encrypt your log files with your own KMS key.

   Choose **Existing** to update your event data store with your KMS key. Choose a KMS key that is in the same Region as the event data store. A key from another account is not supported.

   In **Enter Amazon KMS Alias**, specify the alias for which you changed the policy for use with CloudTrail, in the format `alias/MyAliasName`. For more information, see Updating a resource to use your KMS key (p. 394).

   You can choose an alias, or use the globally unique key ID. The value can be one of the following formats:

   - **Alias Name**: `alias/MyAliasName`
   - **Alias ARN**: `arn:aws:kms:region:123456789012:alias/MyAliasName`
   - **Key ARN**: `arn:aws:kms:region:123456789012:key/12345678-1234-1234-1234-123456789012`
   - **Globally unique key ID**: `12345678-1234-1234-1234-123456789012`
5. Choose **Save changes**.

   Note
   If the KMS key that you chose is disabled or is pending deletion, you cannot save the event data store configuration with that KMS key. You can enable the KMS key, or choose a different key. For more information, see Key state: Effect on your KMS key in the *Amazon Key Management Service Developer Guide*.

Enabling and disabling CloudTrail log file encryption with the Amazon CLI

This topic describes how to enable and disable SSE-KMS log file encryption for CloudTrail by using the Amazon CLI. For background information, see Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386).

Topics

- Enabling CloudTrail log file encryption by using the Amazon CLI (p. 396)
- Disabling CloudTrail log file encryption by using the Amazon CLI (p. 398)

Enabling CloudTrail log file encryption by using the Amazon CLI

- Enable log file encryption for a trail (p. 397)

- Enable log file encryption for an event data store (p. 397)

Enable log file encryption for a trail

1. Create a key with the Amazon CLI. The key that you create must be in the same Region as the S3 bucket that receives your CloudTrail log files. For this step, you use the Amazon KMS **create-key** command.
2. Get the existing key policy so that you can modify it for use with CloudTrail. You can retrieve the key policy with the Amazon KMS **get-key-policy** command.
3. Add required sections to the key policy so that CloudTrail can encrypt and users can decrypt your log files. Be sure that all users who read the log files are granted decrypt permissions. Do not change existing sections of the policy. For information about the policy sections to include, see Configure Amazon KMS key policies for CloudTrail (p. 387).
4. Attach the modified JSON policy file to the key by using the Amazon KMS **put-key-policy** command.
5. Run the CloudTrail `create-trail` or `update-trail` command with the `--kms-key-id` parameter. This command enables log encryption.

```
aws cloudtrail update-trail --name Default --kms-key-id alias/MyKmsKey
```

The `--kms-key-id` parameter specifies the key whose policy you modified for CloudTrail. It can be any one of the following formats:

- **Alias Name**. Example: `alias/MyAliasName`
- **Alias ARN**. Example: `arn:aws:kms:us-east-2:123456789012:alias/MyAliasName`
- **Key ARN**. Example: `arn:aws:kms:us-east-2:123456789012:key/12345678-1234-1234-1234-123456789012`
- **Globally unique key ID.** Example: `12345678-1234-1234-1234-123456789012`

The following is an example response:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Default",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/Default",
    "LogFileValidationEnabled": false,
    "KmsKeyId": "arn:aws:kms:us-
east-2:123456789012:key/12345678-1234-1234-1234-123456789012",
    "S3BucketName": "my-bucket-name"
}
```

The presence of the `KmsKeyId` element indicates that log file encryption has been enabled. The encrypted log files should appear in your bucket in about 5 minutes.

Enable log file encryption for an event data store

1. Create a key with the Amazon CLI. The key that you create must be in the same Region as the event data store. For this step, run the Amazon KMS **create-key** command.
2. Get the existing key policy to edit for use with CloudTrail. You can get the key policy by running the Amazon KMS **get-key-policy** command.
3. Add required sections to the key policy so that CloudTrail can encrypt and users can decrypt your log files. Be sure that all users who read the log files are granted decrypt permissions. Do not change

existing sections of the policy. For information about the policy sections to include, see Configure Amazon KMS key policies for CloudTrail (p. 387).

4. Attach the edited JSON policy file to the key by running the Amazon KMS put-key-policy command.
5. Run the CloudTrail `create-event-data-store` or `update-event-data-store` command, and add the `--kms-key-id` parameter. This command enables log encryption.

```
aws cloudtrail update-event-data-store --name my-event-data-store --kms-key-id
 alias/MyKmsKey
```

The `--kms-key-id` parameter specifies the key whose policy you modified for CloudTrail. It can be any one of the following four formats:

- **Alias Name**. Example: `alias/MyAliasName`
- **Alias ARN**. Example: `arn:aws:kms:us-east-2:123456789012:alias/MyAliasName`
- **Key ARN**. Example: `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- **Globally unique key ID.** Example: `12345678-1234-1234-1234-123456789012`

The following is an example response:

```
{
    "Name": "my-event-data-store",
    "ARN": "arn:aws:cloudtrail:us-east-1:12345678910:eventdatastore/
EXAMPLEf852-4e8f-8bd1-bcf6cEXAMPLE",
    "RetentionPeriod": "90",
    "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
    "MultiRegionEnabled": false,
    "OrganizationEnabled": false,
    "TerminationProtectionEnabled": true,
    "AdvancedEventSelectors": [{
        "Name": "Select all external events",
        "FieldSelectors": [{
            "Field": "eventCategory",
            "Equals": [
                "ActivityAuditLog"
            ]
        }]
    }]
}
```

The presence of the `KmsKeyId` element indicates that log file encryption has been enabled. The encrypted log files should appear in your event data store in about 5 minutes.

Disabling CloudTrail log file encryption by using the Amazon CLI

To stop encrypting logs on a trail, run `update-trail` and pass an empty string to the `kms-key-id` parameter:

```
aws cloudtrail update-trail --name my-test-trail --kms-key-id ""
```

The following is an example response:

```
{
    "IncludeGlobalServiceEvents": true,
```

```
    "Name": "Default",
    "TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/Default",
    "LogFileValidationEnabled": false,
    "S3BucketName": "my-bucket-name"
}
```

The absence of the `KmsKeyId` value indicates that log file encryption is no longer enabled.

Important
You cannot stop log file encryption on an event data store.

CloudTrail log event reference

A CloudTrail log is a record in JSON format. The log contains information about requests for resources in your account, such as who made the request, the services used, the actions performed, and parameters for the action. The event data is enclosed in a `Records` array.

The following example shows a single log record of an event. In this event, an IAM user named `Mary_Major` called the CloudTrail `StartLogging` API from the CloudTrail console to start the logging process.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Mary_Major",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Mary_Major",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2019-06-18T22:28:31Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2019-06-19T00:18:31Z",
    "eventSource": "cloudtrail.amazonaws.com",
    "eventName": "StartLogging",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "203.0.113.64",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "name": "arn:aws:cloudtrail:us-east-2:123456789012:trail/My-First-Trail"
    },
    "responseElements": null,
    "requestID": "ddf5140f-EXAMPLE",
    "eventID": "7116c6a1-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
},
    ... additional entries ...
```

There are two events logged to show unusual activity in CloudTrail Insights: a start event and an end event. The following example shows a single log record of a starting Insights event that occurred when the Application Auto Scaling API `CompleteLifecycleAction` was called an unusual number of times. For Insights events, the value of `eventCategory` is `Insight`. An `insightDetails` block identifies the event state, source, name, Insights type, and context, including statistics and attributions. For more information about the `insightDetails` block, see CloudTrail Insights `insightDetails` element (p. 424).

```
{
          "eventVersion": "1.07",
          "eventTime": "2020-07-21T20:56:00Z",
          "awsRegion": "us-east-1",
          "eventID": "abcd00b0-ccfe-422d-961c-98a2198a408x",
```

```
"eventType": "AwsCloudTrailInsight",
"recipientAccountId": "838185438692",
"sharedEventID": "7bb000gg-22b3-4c03-94af-c74tj0c8m7c0",
"insightDetails": {
  "state": "Start",
  "eventSource": "autoscaling.amazonaws.com",
  "eventName": "CompleteLifecycleAction",
  "insightType": "ApiCallRateInsight",
  "insightContext": {
    "statistics": {
      "baseline": {
        "average": 0.0000882145
      },
      "insight": {
        "average": 0.6
      },
      "insightDuration": 5,
      "baselineDuration": 11336
    },
    "attributions": [
      {
        "attribute": "userIdentityArn",
        "insight": [
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
            "average": 0.2
          },
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole2",
            "average": 0.2
          },
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole3",
            "average": 0.2
          }
        ],
        "baseline": [
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
            "average": 0.0000882145
          }
        ]
      },
      {
        "attribute": "userAgent",
        "insight": [
          {
            "value": "codedeploy.amazonaws.com",
            "average": 0.6
          }
        ],
        "baseline": [
          {
            "value": "codedeploy.amazonaws.com",
            "average": 0.0000882145
          }
        ]
      },
      {
        "attribute": "errorCode",
        "insight": [
          {
            "value": "null",
            "average": 0.6
          }
        ],
```

```
                "baseline": [
                  {
                    "value": "null",
                    "average": 0.0000882145
                  }
                ]
              }
            ]
          }
        },
        "eventCategory": "Insight"
}
```

The following topics list the data fields that CloudTrail captures for each Amazon API call and sign-in event.

Topics

- CloudTrail record contents (p. 402)
- CloudTrail userIdentity element (p. 416)
- CloudTrail Insights insightDetails element (p. 424)
- Non-API events captured by CloudTrail (p. 430)

CloudTrail record contents

The body of the record contains fields that help you determine the requested action as well as when and where the request was made. When the value of **Optional** is **True**, the field is only present when it applies to the service, API, or event type. An **Optional** value of **False** means that the field is either always present, or that its presence does not depend on the service, API, or event type. An example is `responseElements`, which is present in events for actions that make changes (create, update, or delete actions).

`eventTime`

The date and time the request was completed, in coordinated universal time (UTC). An event's time stamp comes from the local host that provides the service API endpoint on which the API call was made. For example, a **CreateBucket** API event that is run in the US West (Oregon) Region would get its time stamp from the time on an Amazon host running the Amazon S3 endpoint, `s3.us-west-2.amazonaws.com`. In general, Amazon services use Network Time Protocol (NTP) to synchronize their system clocks.

Since: 1.0

Optional: False

`eventVersion`

The version of the log event format. The current version is 1.09.

The `eventVersion` value is a major and minor version in the form *`major_version.minor_version`*. For example, you can have an `eventVersion` value of `1.08`, where `1` is the major version, and `08` is the minor version.

CloudTrail increments the major version if a change is made to the event structure that is not backward-compatible. This includes removing a JSON field that already exists, or changing how the contents of a field are represented (for example, a date format). CloudTrail increments the minor version if a change adds new fields to the event structure. This can occur if new information is available for some or all existing events, or if new information is available only for new event types. Applications can ignore new fields to stay compatible with new minor versions of the event structure.

If CloudTrail introduces new event types, but the structure of the event is otherwise unchanged, the event version does not change.

To be sure that your applications can parse the event structure, we recommend that you perform an equal-to comparison on the major version number. To be sure that fields that are expected by your application exist, we also recommend performing a greater-than-or-equal-to comparison on the minor version. There are no leading zeroes in the minor version. You can interpret both *major_version* and *minor_version* as numbers, and perform comparison operations.

Since: 1.0

Optional: False

userIdentity

Information about the IAM identity that made a request. For more information, see CloudTrail userIdentity element (p. 416).

Since: 1.0

Optional: False

eventSource

The service that the request was made to. This name is typically a short form of the service name without spaces plus `.amazonaws.com`. For example:

- Amazon CloudFormation is `cloudformation.amazonaws.com`.
- Amazon EC2 is `ec2.amazonaws.com`.
- Amazon Simple Workflow Service is `swf.amazonaws.com`.

This convention has some exceptions. For example, the `eventSource` for Amazon CloudWatch is `monitoring.amazonaws.com`.

Since: 1.0

Optional: False

eventName

The requested action, which is one of the actions in the API for that service.

Since: 1.0

Optional: False

awsRegion

The Amazon Web Services Region that the request was made to, such as `us-east-2`. See CloudTrail supported Regions (p. 15).

Since: 1.0

Optional: False

sourceIPAddress

The IP address that the request was made from. For actions that originate from the service console, the address reported is for the underlying customer resource, not the console web server. For services in Amazon, only the DNS name is displayed.

Since: 1.0

Optional: False

> **Note**
> For events originated by Amazon, this field is usually `AWS Internal/#`, where `#` is a number used for internal purposes.

`userAgent`

The agent through which the request was made, such as the Amazon Web Services Management Console, an Amazon service, the Amazon SDKs or the Amazon CLI. This field has a maximum size of 1 KB; content exceeding that limit is truncated. The following are example values:

- `signin.amazonaws.com` – The request was made by an IAM user with the Amazon Web Services Management Console.
- `console.amazonaws.com` – The request was made by a root user with the Amazon Web Services Management Console.
- `lambda.amazonaws.com` – The request was made with Amazon Lambda.
- `aws-sdk-java` – The request was made with the Amazon SDK for Java.
- `aws-sdk-ruby` – The request was made with the Amazon SDK for Ruby.
- `aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5` – The request was made with the Amazon CLI installed on Linux.

> **Note**
> For events originated by Amazon, this field is usually `AWS Internal/#`, where `#` is a number used for internal purposes.

Since: 1.0

Optional: True

`errorCode`

The Amazon service error if the request returns an error. For an example that shows this field, see Error code and message log example (p. 22). This field has a maximum size of 1 KB; content exceeding that limit is truncated.

Since: 1.0

Optional: True

`errorMessage`

If the request returns an error, the description of the error. This message includes messages for authorization failures. CloudTrail captures the message logged by the service in its exception handling. For an example, see Error code and message log example (p. 22). This field has a maximum size of 1 KB; content exceeding that limit is truncated.

> **Note**
> Some Amazon services provide the `errorCode` and `errorMessage` as top-level fields in the event. Other Amazon services provide error information as part of `responseElements`.

Since: 1.0

Optional: True

`requestParameters`

The parameters, if any, that were sent with the request. These parameters are documented in the API reference documentation for the appropriate Amazon service. This field has a maximum size of 100 KB; content exceeding that limit is truncated.

Since: 1.0

Optional: False

`responseElements`

The response element for actions that make changes (create, update, or delete actions). If an action does not change state (for example, a request to get or list objects), this element is omitted. These actions are documented in the API reference documentation for the appropriate Amazon service. This field has a maximum size of 100 KB; content exceeding that limit is truncated.

The `responseElements` value is useful to help you trace a request with Amazon Web Services Support. Both `x-amz-request-id` and `x-amz-id-2` contain information that helps you trace a request with Amazon Web Services Support. These values are the same as those that the service returns in the response to the request that initiates the events, so you can use them to match the event to the request.

Since: 1.0

Optional: False

`additionalEventData`

Additional data about the event that was not part of the request or response. This field has a maximum size of 28 KB; content exceeding that limit is truncated.

Support for this field begins with `eventVersion` `1.00`.

Since: 1.0

Optional: True

`requestID`

The value that identifies the request. The service being called generates this value. This field has a maximum size of 1 KB; content exceeding that limit is truncated.

Support for this field begins with `eventVersion` `1.01`.

Since: 1.01

Optional: False

`eventID`

GUID generated by CloudTrail to uniquely identify each event. You can use this value to identify a single event. For example, you can use the ID as a primary key to retrieve log data from a searchable database.

Since: 1.01

Optional: False

`eventType`

Identifies the type of event that generated the event record. This can be the one of the following values:

- `AwsApiCall` – An API was called.
- `AwsServiceEvent (p. 430)` – The service generated an event related to your trail. For example, this can occur when another account made a call with a resource that you own.
- `AwsConsoleAction` – An action was taken in the console that was not an API call.
- `AwsConsoleSignIn (p. 431)` – A user in your account (root, IAM, federated, SAML, or SwitchRole) signed in to the Amazon Web Services Management Console.
- `AwsCloudTrailInsight (p. 267)` – If Insights events are enabled for the trail, CloudTrail generates Insights events when CloudTrail detects unusual operational activity such as spikes in resource provisioning or bursts of Amazon Identity and Access Management (IAM) actions.

  `AwsCloudTrailInsight` events do *not* use the following fields:
  - `eventName`
  - `eventSource`
  - `sourceIPAddress`
  - `userAgent`
  - `userIdentity`

Since: 1.02

Optional: False

`apiVersion`

Identifies the API version associated with the `AwsApiCall` `eventType` value.

Since: 1.01

Optional: True

`managementEvent`

A Boolean value that identifies whether the event is a management event. `managementEvent` is shown in an event record if `eventVersion` is 1.06 or higher, and the event type is one of the following:

- `AwsApiCall`
- `AwsConsoleAction`
- `AwsConsoleSignIn`
- `AwsServiceEvent`

Since: 1.06

Optional: True

`readOnly`

Identifies whether this operation is a read-only operation. This can be one of the following values:

- `true` – The operation is read-only (for example, `DescribeTrails`).
- `false` – The operation is write-only (for example, `DeleteTrail`).

Since: 1.01

Optional: True

`resources`

A list of resources accessed in the event. The field can contain the following information:

- Resource ARNs
- Account ID of the resource owner
- Resource type identifier in the format: `AWS::`*`aws-service-name`*`::`*`data-type-name`*

For example, when an `AssumeRole` event is logged, the `resources` field can appear like the following:

- ARN: `arn:aws:iam::123456789012:role/`*`myRole`*
- Account ID: `123456789012`
- Resource type identifier: `AWS::IAM::Role`

For example logs with the `resources` field, see Amazon STS API Event in CloudTrail Log File in the *IAM User Guide* or Logging Amazon KMS API Calls in the *Amazon Key Management Service Developer Guide*.

Since: 1.01

Optional: True

`recipientAccountId`

Represents the account ID that received this event. The `recipientAccountID` may be different from the CloudTrail userIdentity element (p. 416) `accountId`. This can occur in cross-account resource access. For example, if a KMS key, also known as an Amazon KMS key, was used by a separate account to call the Encrypt API, the `accountId` and `recipientAccountID` values will be the same for the event delivered to the account that made the call, but the values will be different for the event that is delivered to the account that owns the KMS key.

Since: 1.02

Optional: True

`serviceEventDetails`

Identifies the service event, including what triggered the event and the result. For more information, see Amazon service events (p. 430). This field has a maximum size of 100 KB; content exceeding that limit is truncated.

Since: 1.05

Optional: True

`sharedEventID`

GUID generated by CloudTrail to uniquely identify CloudTrail events from the same Amazon action that is sent to different Amazon accounts.

For example, when an account uses an Amazon KMS key that belongs to another account, the account that used the KMS key and the account that owns the KMS key receive separate CloudTrail events for the same action. Each CloudTrail event delivered for this Amazon action shares the same `sharedEventID`, but also has a unique `eventID` and `recipientAccountID`.

For more information, see Example sharedEventID (p. 410).

Note
The `sharedEventID` field is present only when CloudTrail events are delivered to multiple accounts. If the caller and owner are the same Amazon account, CloudTrail sends only one event, and the `sharedEventID` field is not present.

Since: 1.03

Optional: True

`vpcEndpointId`

Identifies the VPC endpoint in which requests were made from a VPC to another Amazon service, such as Amazon S3.

Since: 1.04

Optional: True

`eventCategory`

Shows the event category that is used in `LookupEvents` calls.

- For management events, the value is `Management`.
- For data events, the value is `Data`.
- For Insights events, the value is `Insight`.

Since: 1.07

Optional: False

`addendum`

If an event delivery was delayed, or additional information about an existing event becomes available after the event is logged, an addendum field shows information about why the event was delayed. If information was missing from an existing event, the addendum field includes the missing information and a reason for why it was missing. Contents include the following.

- **`reason`** - The reason that the event or some of its contents were missing. Values can be any of the following.
  - **`DELIVERY_DELAY`** – There was a delay delivering events. This could be caused by high network traffic, connectivity issues, or a CloudTrail service issue.
  - **`UPDATED_DATA`** – A field in the event record was missing or had an incorrect value.
  - **`SERVICE_OUTAGE`** – A service that logs events to CloudTrail had an outage, and couldn't log events to CloudTrail. This is exceptionally rare.
- **`updatedFields`** - The event record fields that are updated by the addendum. This is only provided if the reason is `UPDATED_DATA`.
- **`originalRequestID`** - The original unique ID of the request. This is only provided if the reason is `UPDATED_DATA`.
- **`originalEventID`** - The original event ID. This is only provided if the reason is `UPDATED_DATA`.

Since: 1.08

Optional: True

`sessionCredentialFromConsole`

Shows whether or not an event originated from an Amazon Web Services Management Console session. This field is not shown unless the value is `true`, meaning that the client that was used to make the API call was either a proxy or an external client. If a proxy client was used, the`tlsDetails` event field is not shown.

Since: 1.08

Optional: True

`edgeDeviceDetails`

Shows information about edge devices that are targets of a request. Currently, `S3 Outposts` device events include this field. This field has a maximum size of 28 KB; content exceeding that limit is truncated.

Since: 1.08

Optional: True

`tlsDetails`

Shows information about the Transport Layer Security (TLS) version, cipher suites, and the fully qualified domain name (FQDN) of the client-provided host name used in the service API call, which is typically the FQDN of the service endpoint. CloudTrail still logs partial TLS details if expected information is missing or empty. For example, if the TLS version and cipher suite are present, but the `HOST` header is empty, available TLS details are still logged in the CloudTrail event. For more information about which services log TLS details to CloudTrail, see Services that support TLS details in CloudTrail (p. 411).

If `sessionCredentialFromConsole` is present with a value of `true`, `tlsDetails` is present in an event record only if an external client was used to make the API call.

- **`tlsVersion`** - The TLS version of a request.
- **`cipherSuite`** - The cipher suite (combination of security algorithms used) of a request.
- **`clientProvidedHostHeader`** - The client-provided host name used in the service API call, which is typically the FQDN of the service endpoint.

Since: 1.08

Optional: True

Record fields for Insights events

The following are attributes shown in the JSON structure of an Insights event that differ from those in a management or data event.

`sharedEventId`

A `sharedEventID` for CloudTrail Insights events differs from the `sharedEventID` for the management and data types of CloudTrail events. In Insights events, a `sharedEventID` is a GUID that is generated by CloudTrail Insights to uniquely identify an Insights event. `sharedEventID` is common between the start and the end Insights events, and helps to connect both events to uniquely identify unusual activity. You can think of the `sharedEventID` as the overall Insights event ID.

Since: 1.07

Optional: False

`insightDetails`

Insights events only. Shows information about the underlying triggers of an Insights event, such as event source, user agent, statistics, API name, and whether the event is the start or end of the Insights event. For more information about the contents of the `insightDetails` block, see CloudTrail Insights `insightDetails` element (p. 424).

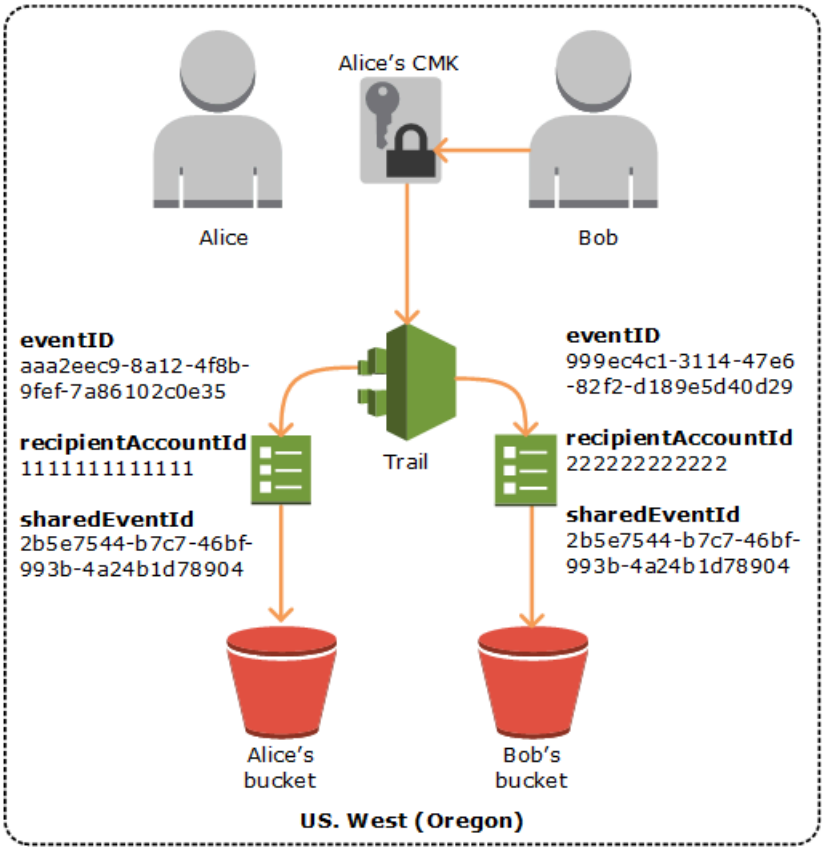
Since: 1.07

Optional: False

Example sharedEventID

The following is an example that describes how CloudTrail delivers two events for the same action:

1. Alice has Amazon account (111111111111) and creates an Amazon KMS key. She is the owner of this KMS key.
2. Bob has Amazon account (222222222222). Alice gives Bob permission to use the KMS key.
3. Each account has a trail and a separate bucket.
4. Bob uses the KMS key to call the `Encrypt` API.
5. CloudTrail sends two separate events.
   - One event is sent to Bob. The event shows that he used the KMS key.
   - One event is sent to Alice. The event shows that Bob used the KMS key.
   - The events have the same `sharedEventID`, but the `eventID` and `recipientAccountID` are unique.



Shared event IDs in CloudTrail Insights

A `sharedEventID` for CloudTrail Insights events differs from the `sharedEventID` for the management and data types of CloudTrail events. In Insights events, a `sharedEventID` is a GUID that is generated

by CloudTrail Insights to uniquely identify a start and end pair of Insights events. `sharedEventID` is common between the start and the end Insights event, and helps to create a correlation between both events to uniquely identify unusual activity.

You can think of the `sharedEventID` as the overall Insights event ID.

Services that support TLS details in CloudTrail

Starting on June 28, 2023, Amazon requires the Transport Layer Security (TLS) configuration for all Amazon service API endpoints to have a minimum version of TLS 1.2. For more information, see the blog post, TLS 1.2 to become the minimum TLS protocol level for all Amazon API endpoints. The `tlsDetails` structure in each CloudTrail record contains the TLS version, cipher suite, and the client-provided host name used in the service API call, which is typically the fully qualified domain name (FQDN) of the service endpoint. You can then use the data in the records to help you pinpoint your client software that is using an older TLS version, and update it accordingly. Nearly half of Amazon services currently provide the TLS information in the CloudTrail `tlsDetails` field. The following table shows Amazon services that display TLS information in CloudTrail records.

Services that support TLS details
Alexa for Business
Amazon Activate
Amazon AppConfig
Amazon App Mesh
Amazon App Runner
Amazon AppStream 2.0
Amazon Auto Scaling
Amazon Backup
Amazon Backup Gateway
Amazon Billing
Amazon Certificate Manager
Amazon Cloud9
Amazon Cloud Directory
Amazon CloudFormation
Amazon CloudFront
Amazon Cloud Map
Amazon CloudSearch
Amazon CloudTrail
Amazon CloudWatch
Amazon CloudWatch Application Insights
Amazon CloudWatch Events

Services that support TLS details
Amazon CloudWatch Logs
Amazon CodeArtifact
Amazon CodeBuild
Amazon CodeCommit
Amazon CodeDeploy
Amazon CodePipeline
AWS CodeStar
AWS CodeStar Connections
Amazon Comprehend
Amazon Comprehend Medical
Amazon Compute Optimizer
Amazon Connect Voice ID
Amazon Control Tower
Amazon Cost and Usage Report
Amazon Cost Explorer
Amazon Database Migration Service (DMS)
Amazon Data Pipeline
Amazon DataSync
Amazon DeepRacer
Amazon Device Farm
Amazon Diode
Amazon Direct Connect
Amazon Directory Service
Amazon DynamoDB
Amazon DynamoDB Accelerator (DAX)
Amazon Elastic Block Store (EBS)
Amazon Elastic Compute Cloud (EC2)
Amazon EC2 Instance Connect
Amazon Elastic Container Registry (ECR)
Amazon Elastic Container Registry (ECR) Public
Amazon Elastic Container Service (ECS)

Services that support TLS details
Amazon ElastiCache
Amazon Elastic File System (EFS)
Amazon Elastic Transcoder
Amazon Elastic Load Balancing (ELB)
Amazon Elastic Load Balancing (ELBV2)
AWS Elemental MediaStore
Amazon EMR
Amazon EventBridge
Amazon Firewall Manager
Amazon Forecast
Amazon Fraud Detector
Amazon FSx
Amazon GameLift
Amazon Global Accelerator
Amazon Glue (Lake Formation)
Amazon HealthLake
Amazon Identity and Access Management (IAM)
Amazon Identity Store
Amazon Inspector
Amazon IoT Analytics
Amazon IoT Core
Amazon IoT Events
Amazon IoT Secure Tunneling
Amazon IoT SiteWise
Amazon IoT Wireless
Amazon Kendra
Amazon Key Management Service (KMS)
Amazon Kinesis
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Kinesis Data Streams

Services that support TLS details
Amazon Kinesis Video Streams
Amazon Lambda
Amazon License Manager
Amazon Lightsail
Amazon Lookout for Equipment
Amazon Machine Learning
Amazon Managed Service for Prometheus
Amazon Managed Services
Amazon Marketplace Commerce Analytics
Amazon Web Services Marketplace Discovery
Amazon Marketplace Entitlement Service
Amazon Marketplace Metering Service
Amazon Mechanical Turk
Amazon MemoryDB for Redis
Amazon Migration Hub
Amazon Network Firewall
Amazon OpenSearch Service
Amazon OpsWorks CM
Amazon Organizations
Amazon Polly
Amazon Price List
Amazon Private Certificate Authority
Amazon Proton
Amazon QuickSight
Amazon Redshift
Amazon Rekognition
Amazon Relational Database Service (RDS)
Amazon Relational Database Service (RDS) Data API
Amazon Resource Groups Tagging
Amazon Route 53
Amazon Route 53 Domains

Services that support TLS details
Amazon Route 53 Resolver
Amazon SageMaker
Amazon SageMaker-Edge
Amazon Secrets Manager
Amazon Security Token Service (STS)
Amazon Service Catalog
Amazon Service Quotas
Amazon Shield
Amazon SimpleDB
Amazon Simple Email Service (SES)
Amazon Simple Notification Service (SNS)
Amazon Simple Queue Service (SQS)
Amazon Simple Storage Service (S3)
Amazon S3 Glacier
Amazon Simple Workflow Service (SWF)
Amazon Snowball
Amazon Step Functions
Amazon Storage Gateway
Amazon Web Services Support
Amazon Systems Manager
Amazon Textract
Amazon Timestream
Amazon Transcribe Streaming Service
Amazon Transfer Family
Amazon Translate
Amazon Trusted Advisor
Amazon WAF
Amazon WorkDocs
Amazon WorkMail
Amazon WorkMail Message Flow
Amazon WorkSpaces

Services that support TLS details
Amazon X-Ray

CloudTrail userIdentity element

Amazon Identity and Access Management (IAM) provides different types of identities. The `userIdentity` element contains details about the type of IAM identity that made the request, and which credentials were used. If temporary credentials were used, the element shows how the credentials were obtained.

Contents

- Examples (p. 416)
- Fields (p. 417)
- Values for Amazon STS APIs with SAML and web identity federation (p. 421)
- Amazon STS source identity (p. 422)

Examples

`userIdentity` with IAM user credentials

The following example shows the `userIdentity` element of a simple request made with the credentials of the IAM user named `Alice`.

```
"userIdentity": {
  "type": "IAMUser",
  "principalId": "AIDAJ45Q7YFFAREXAMPLE",
  "arn": "arn:aws:iam::123456789012:user/Alice",
  "accountId": "123456789012",
  "accessKeyId": "",
  "userName": "Alice"
}
```

`userIdentity` with temporary security credentials

The following example shows a `userIdentity` element for a request made with temporary security credentials obtained by assuming an IAM role. The element contains additional details about the role that was assumed to get credentials.

```
"userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
    "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/MySessionName",
    "accountId": "123456789012",
    "accessKeyId": "",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "20131102T010628Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIDPPEZS35WEXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/RoleToBeAssumed",
        "accountId": "123456789012",
```

```
        "userName": "RoleToBeAssumed"
      }
    }
}
```

`userIdentity` for a request made on behalf of an IAM Identity Center user

The following example shows a `userIdentity` element for a request made on behalf of an IAM Identity Center user.

```
"userIdentity": {
 "type": "IdentityCenterUser",
 "accountId": "123456789012",
 "onBehalfOf": {
  "userId": "544894e8-80c1-707f-60e3-3ba6510dfac1",
  "identityStoreArn": "arn:aws:identitystore::123456789012:identitystore/d-9067642ac7"
 },
 "credentialId": "EXAMPLEVHULjJdTUdPJfofVa1sufHDoj7aYcOYcxFVllWR_Whr1fEXAMPLE"
}
```

Fields

The following fields can appear in a `userIdentity` element.

`type`

The type of the identity. The following values are possible:

- `Root` – The request was made with your Amazon Web Services account credentials. If the `userIdentity` type is `Root`, and you set an alias for your account, the `userName` field contains your account alias. For more information, see Your Amazon Web Services account ID and its alias.
- `IAMUser` – The request was made with the credentials of an IAM user.
- `AssumedRole` – The request was made with temporary security credentials that were obtained with a role by making a call to the Amazon Security Token Service (Amazon STS) `AssumeRole` API. This can include roles for Amazon EC2 and cross-account API access.
- `Role` – The request was made with a persistent IAM identity that has specific permissions. The issuer of the role sessions is always the role. For more information about roles, see Roles terms and concepts in the *IAM User Guide*.
- `FederatedUser` – The request was made with temporary security credentials obtained from a call to the Amazon STS `GetFederationToken` API. The `sessionIssuer` element indicates if the API was called with root or IAM user credentials.

  For more information about temporary security credentials, see Temporary Security Credentials in the *IAM User Guide*.
- `Directory` – The request was made to a directory service, and the type is unknown. Directory services include the following: Amazon WorkDocs and Amazon QuickSight.
- `AWSAccount` – The request was made by another Amazon Web Services account
- `AWSService` – The request was made by an Amazon Web Services account that belongs to an Amazon Web Service. For example, Amazon Elastic Beanstalk assumes an IAM role in your account to call other Amazon Web Services on your behalf.
- `IdentityCenterUser` – The request was made on behalf of an IAM Identity Center user.
- Unknown – The request was made with an identity type that CloudTrail can't determine.

Optional: False

`AWSAccount` and `AWSService` appear for `type` in your logs when there is cross-account access using an IAM role that you own.

Example: Cross-account access initiated by another Amazon account

1. You own an IAM role in your account.
2. Another Amazon account switches to that role to assume the role for your account.
3. Because you own the IAM role, you receive a log that shows the other account assumed the role. The `type` is `AWSAccount`. For an example log entry, see Amazon STS API event in CloudTrail log file.

Example: Cross-account access initiated by an Amazon service

1. You own an IAM role in your account.
2. An Amazon account owned by an Amazon service assumes that role.
3. Because you own the IAM role, you receive a log that shows the Amazon service assumed the role. The `type` is `AWSService`.

`userName`

The friendly name of the identity that made the call. The value that appears in `userName` is based on the value in `type`. The following table shows the relationship between `type` and `userName`:

type	userName	Description
`Root` (no alias set)	Not present	If you haven't set up an alias for your Amazon Web Services account, the `userName` field doesn't appear. For more information about account aliases, see Your Amazon Web Services account ID and its alias. Note that the `userName` field can't contain `Root`, because `Root` is an identity type and not a user name.
`Root` (alias set)	The account alias	For more information about Amazon Web Services account aliases, see Your Amazon Web Services account ID and its alias.
`IAMUser`	The user name of the IAM user	
`AssumedRole`	Not present	For the `AssumedRole` type, you can find the `userName` field in `sessionContext` as part of the **`sessionIssuer`** (p. 420) element. For an example entry, see Examples (p. 416).
`Role`	User-defined	The `sessionContext` and `sessionIssuer` section contains information about the identity that issued the session for the role.
`FederatedUser`	Not present	The `sessionContext` and `sessionIssuer` section contains information about the identity that issued the session for the federated user.
`Directory`	Can be present	For example, the value can be the account alias or email address of the associated Amazon Web Services account ID.
`AWSService`	Not present	

type	userName	Description
AWSAccount	Not present	
IdentityCenterUser	Not present	The onBehalfOf section contains information about the IAM Identity Center user ID and identity store ARN for which the call was made. For more information about IAM Identity Center, see the *Amazon IAM Identity Center User Guide*.
Unknown	Can be present	For example, the value can be the account alias or email address of the associated Amazon Web Services account ID.

Optional: True

Note
The userName field contains the string HIDDEN_DUE_TO_SECURITY_REASONS when the recorded event is a console sign-in failure caused by incorrect user name input. CloudTrail does not record the contents in this case because the text could contain sensitive information, as in the following examples:

- A user accidentally types a password in the user name field.
- A user clicks the link for one Amazon account's sign-in page, but then types the account number for a different one.
- A user accidentally types the account name of a personal email account, a bank sign-in identifier, or some other private ID.

principalId

A unique identifier for the entity that made the call. For requests made with temporary security credentials, this value includes the session name that is passed to the AssumeRole, AssumeRoleWithWebIdentity, or GetFederationToken API call.

Optional: True

arn

The Amazon Resource Name (ARN) of the principal that made the call. The last section of the arn contains the user or role that made the call.

Optional: True

accountId

The account that owns the entity that granted permissions for the request. If the request was made with temporary security credentials, this is the account that owns the IAM user or role used to obtain credentials.

If the request was made with an IAM Identity Center authorized access token, this is the account that owns the IAM Identity Center instance.

Optional: True

accessKeyId

The access key ID that was used to sign the request. If the request was made with temporary security credentials, this is the access key ID of the temporary credentials. For security reasons, accessKeyId might not be present, or might be displayed as an empty string.

Optional: True

`sessionContext`

If the request was made with temporary security credentials, `sessionContext` provides information about the session created for those credentials. You create a session when you call any API that returns temporary credentials. Users also create sessions when they work in the console and make requests with APIs that include multi-factor authentication. This element has the following attributes:

- `creationDate` – The date and time when the temporary security credentials were issued. Represented in ISO 8601 basic notation.
- `mfaAuthenticated` – The value is `true` if the root user or IAM user who used their credentials for the request also authenticated with an MFA device; otherwise, `false`.
- `sourceIdentity` – See Amazon STS source identity (p. 422) in this topic. The `sourceIdentity` field occurs in events when users assume an IAM role to perform an action. `sourceIdentity` identifies the original user identity making the request, whether that user's identity is an IAM user, an IAM role, a user authenticated through SAML-based federation, or a user authenticated through OpenID Connect (OIDC)-compliant web identity federation. For more information about configuring Amazon STS to collect source identity information, see Monitor and control actions taken with assumed roles in the *IAM User Guide*.
- `ec2RoleDelivery` – The value is `1.0` if the credentials were provided by Amazon EC2 Instance Metadata Service Version 1 (IMDSv1). The value is `2.0` if the credentials were provided using the new IMDS scheme.

  Amazon credentials provided by the Amazon EC2 Instance Metadata Service (IMDS) include an ec2:RoleDelivery IAM context key. This context key makes it easy to enforce use of the new scheme on a service-by-service or resource-by-resource basis by using the context key as a condition in IAM policies, resource policies, or Amazon Organizations service control policies. For more information, see Instance metadata and user data in the *Amazon EC2 User Guide for Linux Instances*.

Optional: True

`invokedBy`

The name of the Amazon Web Service that made the request, such as Amazon EC2 Auto Scaling or Amazon Elastic Beanstalk.

Optional: True

`sessionIssuer`

If a user make a request with temporary security credentials, `sessionIssuer` provides information about how the user obtained credentials. For example, if the they obtained temporary security credentials by assuming a role, this element provides information about the assumed role. If they obtained credentials with root or IAM user credentials to call Amazon STS `GetFederationToken`, the element provides information about the root account or IAM user. This element has the following attributes:

- `type` – The source of the temporary security credentials, such as `Root`, `IAMUser`, or `Role`.
- `userName` – The friendly name of the user or role that issued the session. The value that appears depends on the `sessionIssuer` identity `type`. The following table shows the relationship between `sessionIssuer type` and `userName`:

`sessionIssuer` type	**`userName`**	**Description**
`Root` (no alias set)	Not present	If you have not set up an alias for your account, the `userName` field does not appear. For more information about Amazon Web Services account aliases, see Your Amazon Web Services account ID

sessionIssuer type	userName	Description
		and its alias. Note that the userName field can't contain Root, because Root is an identity type, not a user name.
Root (alias set)	The account alias	For more information about Amazon Web Services account aliases, see Your Amazon account ID and its alias.
IAMUser	The user name of the IAM user	This also applies when a federated user is using a session issued by IAMUser.
Role	The role name	A role assumed by an IAM user, Amazon Web Service, or web identity federated user in a role session.

- principalId – The internal ID of the entity used to get credentials.
- arn – The ARN of the source (account, IAM user, or role) that was used to get temporary security credentials.
- accountId – The account that owns the entity that was used to get credentials.

Optional: True

onBehalfOf

If the request was made by an IAM Identity Center caller, onBehalfOf provides information about the IAM Identity Center user ID and identity store ARN for which the call was made. This element has the following attributes:

- userId – The ID of the IAM Identity Center user who the call was made on behalf of.
- identityStoreArn – The ARN of the IAM Identity Center identity store that the call was made on behalf of.

Optional: True

credentialId

The credential ID for the request. This is only set when the caller uses a bearer token, such as an IAM Identity Center authorized access token.

Optional: True

webIdFederationData

If the request was made with temporary security credentials obtained by web identity federation, webIdFederationData lists information about the identity provider. This element has the following attributes:

- federatedProvider – The principal name of the identity provider (for example, www.amazon.com for Login with Amazon or accounts.google.com for Google).
- attributes – The application ID and user ID as reported by the provider (for example, www.amazon.com:app_id and www.amazon.com:user_id for Login with Amazon).

Optional: True

Values for Amazon STS APIs with SAML and web identity federation

Amazon CloudTrail supports logging Amazon Security Token Service (Amazon STS) API calls made with Security Assertion Markup Language (SAML) and web identity federation. When a user makes a call to

the AssumeRoleWithSAML and AssumeRoleWithWebIdentity APIs, CloudTrail records the call and delivers the event to your Amazon S3 bucket.

The userIdentity element for these APIs contains the following values.

type

The identity type.

- SAMLUser – The request was made with SAML assertion.
- WebIdentityUser – The request was made by a web identity federation provider.

principalId

A unique identifier for the entity that made the call.

- For SAMLUser, this is a combination of the saml:namequalifier and saml:sub keys.
- For WebIdentityUser, this is a combination of the issuer, application ID, and user ID.

userName

The name of the identity that made the call.

- For SAMLUser, this is the saml:sub key.
- For WebIdentityUser, this is the user ID.

identityProvider

The principal name of the external identity provider. This field appears only for SAMLUser or WebIdentityUser types.

- For SAMLUser, this is the saml:namequalifier key for the SAML assertion.
- For WebIdentityUser, this is the issuer name of the web identity federation provider. This can be a provider that you configured, such as the following:
  - cognito-identity.amazon.com for Amazon Cognito
  - www.amazon.com for Login with Amazon
  - accounts.google.com for Google
  - graph.facebook.com for Facebook

The following is an example userIdentity element for the AssumeRoleWithWebIdentity action.

```
"userIdentity": {
    "type": "WebIdentityUser",
    "principalId": "accounts.google.com:application-id.apps.googleusercontent.com:user-id",
    "userName": "user-id",
    "identityProvider": "accounts.google.com"
  }
```

For example logs of how the userIdentity element appears for SAMLUser and WebIdentityUser types, see Logging IAM and Amazon STS API calls with Amazon CloudTrail.

Amazon STS source identity

An IAM administrator can configure Amazon Security Token Service to require that users specify their identity when they use temporary credentials to assume roles. The sourceIdentity field occurs in events when users assume an IAM role or perform any actions with the assumed role.

The `sourceIdentity` field identifies the original user identity making the request, whether that user's identity is an IAM user, an IAM role, a user authenticated by using SAML-based federation, or a user authenticated by using OpenID Connect (OIDC)-compliant web identity federation. After the IAM administrator configures Amazon STS, CloudTrail logs `sourceIdentity` information in the following events and locations within the event record:

- The Amazon STS `AssumeRole`, `AssumeRoleWithSAML`, or `AssumeRoleWithWebIdentity` calls that a user identity makes when it assumes a role. `sourceIdentity` is found in the `requestParameters` block of the Amazon STS calls.
- The Amazon STS `AssumeRole`, `AssumeRoleWithSAML`, or `AssumeRoleWithWebIdentity` calls that a user identity makes if it uses a role to assume another role, known as role chaining. `sourceIdentity` is found in the `requestParameters` block of the Amazon STS calls.
- The Amazon service API calls that the user identity makes while assuming a role and using the temporary credentials assigned by Amazon STS. In service API events, `sourceIdentity` is found in the `sessionContext` block. For example, if a user identity creates a new S3 bucket, `sourceIdentity` occurs in the `sessionContext` block of the `CreateBucket` event.

For more information about how to configure Amazon STS to collect source identity information, see Monitor and control actions taken with assumed roles in the *IAM User Guide*. For more information about Amazon STS events that are logged to CloudTrail, see Logging IAM and Amazon STS API calls with Amazon CloudTrail in the *IAM User Guide*.

The following are example snippets of events that show the `sourceIdentity` field.

Example `requestParameters` section

In the following example event snippet, a user makes an Amazon STS `AssumeRole` request, and sets a source identity, represented here by *`source-identity-value-set`*. The user assumes a role represented by the role ARN `arn:aws:iam::123456789012:role/Assumed_Role`. The `sourceIdentity` field is in the `requestParameters` block of the event.

```
"eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSAccount",
        "principalId": "AIDAJ45Q7YFFAREXAMPLE",
        "accountId": "123456789012"
    },
    "eventTime": "2020-04-02T18:20:53Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.64",
    "userAgent": "aws-cli/1.16.96 Python/3.6.0 Windows/10 botocore/1.12.86",
    "requestParameters": {
        "roleArn": "arn:aws:iam::123456789012:role/Assumed_Role",
        "roleSessionName": "Test1",
        "sourceIdentity": "source-identity-value-set",
    },
```

Example `responseElements` section

In the following example event snippet, a user makes an Amazon STS `AssumeRole` request to assume a role named `Developer_Role`, and sets a source identity, `Admin`. The user assumes a role represented by the role ARN `arn:aws:iam::111122223333:role/Developer_Role`. The `sourceIdentity` field is shown in both the `requestParameters` and `responseElements` blocks of the event. The temporary credentials used to assume the role, the session token string, and the assumed role ID, session name, and session ARN are shown in the `responseElements` block, along with the source identity.

```
    "requestParameters": {
        "roleArn": "arn:aws:iam::111122223333:role/Developer_Role",
        "roleSessionName": "Session_Name",
        "sourceIdentity": "Admin"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "expiration": "Jan 22, 2021 12:46:28 AM",
            "sessionToken": "XXYYaz...
                            EXAMPLE_SESSION_TOKEN
                            XXyYaZAz"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROACKCEVSQ6C2EXAMPLE:Session_Name",
            "arn": "arn:aws:sts::111122223333:assumed-role/Developer_Role/Session_Name"
        },
        "sourceIdentity": "Admin"
    }
...
```

Example `sessionContext` section

In the following example event snippet, a user is assuming a role named `DevRole` to call an Amazon service API. The user sets a source identity, represented here by *`source-identity-value-set`*. The `sourceIdentity` field is in the `sessionContext` block, within the `userIdentity` block of the event.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAJ45Q7YFFAREXAMPLE: Dev1",
    "arn": "arn: aws: sts: : 123456789012: assumed-role/DevRole/Dev1",
    "accountId": "123456789012",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAJ45Q7YFFAREXAMPLE",
        "arn": "arn: aws: iam: : 123456789012: role/DevRole",
        "accountId": "123456789012",
        "userName": "DevRole"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-02-21T23: 46: 28Z"
      },
      "sourceIdentity": "source-identity-value-set"
    }
  }
}
```

CloudTrail Insights `insightDetails` element

Amazon CloudTrail Insights event records include fields that are different from other CloudTrail events in their JSON structure, sometimes called *payload*. A CloudTrail Insights event record includes an `insightDetails` block that contains information about the underlying triggers of an Insights event, such as event source, user identities, user agents, historical averages or *baselines*, statistics, API name,

and whether the event is the start or end of the Insights event. The `insightDetails` block contains the following information.

- **`state`** - Whether the event is the starting or ending Insights event. The value can be `Start` or End.

  Since: 1.07

  Optional: False
- **`eventSource`** - The Amazon service endpoint that was the source of the unusual activity, such as `ec2.amazonaws.com`.

  Since: 1.07

  Optional: False
- **`eventName`** - The name of the Insights event, typically the name of the API that was the source of the unusual activity.

  Since: 1.07

  Optional: False
- **`insightType`** - The type of Insights event. This value can be `ApiCallRateInsight`, `ApiErrorRateInsight`, or both.

  Since: 1.07

  Optional: False
- **`insightContext`** -

  Information about the Amazon tools (called *user agents*), IAM users and roles (called *user identities*), and error codes associated with the events that CloudTrail analyzed to generate the Insights event. This element also includes statistics that show how the unusual activity in an Insights event compares to *baseline*, or normal, activity.

  Since: 1.07

  Optional: False

  - **`statistics`** - Includes data about the *baseline*, or typical average rate of calls to or errors on the subject API by an account as measured during the baseline period, the average rate of calls or errors that triggered the Insights event over the first minute of the Insights event, the duration, in minutes, of the Insights event, and the duration, in minutes, of the baseline measuring period.

    Since: 1.07

    Optional: False

    - **`baseline`** - The average number of API calls or errors per minute during the baseline duration on the Insights event's subject API for the account, calculated over the seven days preceding the start of the Insights event.

      Since: 1.07

      Optional: False
    - **`insight`** -

      For a starting Insights event, this value is the average number of API calls or errors per minute during the start of the unusual activity. For an ending Insights event, this value is the average number of API calls or errors per minute over the duration of the unusual activity.

      Since: 1.07

Optional: False

- **`insightDuration`** - The duration, in minutes, of an Insights event (the time period from the start to the end of unusual activity on the subject API). `insightDuration` occurs in both starting and ending Insights events.

  Since: 1.07

  Optional: False

- **`baselineDuration`** - The duration, in minutes, of the baseline period (the time period that normal activity is measured on the subject API). `baselineDuration` is at minimum the seven days (10080 minutes) preceding an Insights event. This field occurs in both starting and ending Insights events. The ending time of `baselineDuration` measurement is always the start of an Insights event.

  Since: 1.07

  Optional: False

- **`attributions`** - This block includes information about the user identities, user agents, and error codes correlated with unusual and baseline activity. A maximum of five user identities, five user agents, and five error codes are captured in an Insights event `attributions` block, sorted by an average of the count of activity, in descending order from highest to lowest.

  Since: 1.07

  Optional: True

  - **`attribute`** - Contains the attribute type. Value can be `userIdentityArn`, `userAgent`, or `errorCode`.
    - **`userIdentityArn`** - A block that shows up to the top five Amazon users or IAM roles that contributed to API calls or errors during the unusual activity and baseline periods. See also `userIdentity` in CloudTrail record contents (p. 402).

      Since: 1.07

      Optional: False

      - **`insight`** - A block that shows up to the top five user identity ARNs that contributed to the API calls made during the unusual activity period, in descending order from largest number of API calls to smallest. It also shows the average number of API calls made by the user identities during the unusual activity period.

        Since: 1.07

        Optional: False

        - **`value`** - The ARN of one of the top five user identities that contributed to the API calls made during the unusual activity period.

          Since: 1.07

          Optional: False

        - **`average`** - The number of API calls or errors per minute during the unusual activity period for the user identity in the `value` field.

          Since: 1.07

          Optional: False

- **`baseline`** - A block that shows up to the top five user identity ARNs that contributed the most to the API calls or errors during the normal activity period. It also shows the average number of API calls or errors logged by the user identities during the normal activity period.

  Since: 1.07

  Optional: False

  - **`value`** - The ARN of one of the top five user identities that contributed to the API calls or errors during the normal activity period.

    Since: 1.07

    Optional: False

  - **`average`** - The historic average of API calls or errors per minute during the seven days preceding the Insights activity start time for the user identity in the `value` field.

    Since: 1.07

    Optional: False

- **`userAgent`** - A block that shows up to the top five Amazon tools by which the user identity contributed to API calls during the unusual activity and baseline periods. These tools include the Amazon Web Services Management Console, Amazon CLI, or the Amazon SDKs. See also `userAgent` in CloudTrail record contents (p. 402).

  Since: 1.07

  Optional: False

  - **`insight`** - A block that shows up to the top five user agents that contributed to the API calls made during the unusual activity period, in descending order from largest number of API calls to smallest. It also shows the average number of API calls or errors logged by the user agents during the unusual activity period.

    Since: 1.07

    Optional: False

    - **`value`** - One of the top five user agents that contributed to the API calls made during the unusual activity period.

      Since: 1.07

      Optional: False

    - **`average`** - The number of API calls or errors logged per minute during the unusual activity period for the user agent in the `value` field.

      Since: 1.07

      Optional: False

  - **`baseline`** - A block that shows up to the top five user agents that contributed the most to the API calls made during the normal activity period. It also shows the average number of API calls or errors logged by the user agents during the normal activity period.

    Since: 1.07

    Optional: False

    - **`value`** - One of the top five user agents that contributed to the API calls or errors logged during the normal activity period.

Since: 1.07

Optional: False

- **`average`** - The historic average of API calls or errors per minute during the seven days preceding the Insights activity start time for the user agent in the `value` field.

  Since: 1.07

  Optional: False

- **`errorCode`** - A block that shows up to the top five error codes that occurred on API calls during the unusual activity and baseline periods, in descending order from largest number of API calls to smallest. See also `errorCode` in CloudTrail record contents (p. 402).

  Since: 1.07

  Optional: False

  - **`insight`** - A block that shows up to the top five error codes that occurred on the API calls made during the unusual activity period, in descending order from largest number of associated API calls to smallest. It also shows the average number of API calls on which the errors occurred during the unusual activity period.

    Since: 1.07

    Optional: False

    - **`value`** - One of the top five error codes that occurred on the API calls made during the unusual activity period, such as `AccessDeniedException`.

      If none of the calls that triggered the Insights event resulted in errors, this value is `null`.

      Since: 1.07

      Optional: False

    - **`average`** - The number of API calls per minute during the unusual activity period for the error code in the `value` field.

      If the error code value is `null`, and there are no other error codes in the `insight` block, the value of the `average` is the same as that in the `statistics` block for the Insights event overall.

      Since: 1.07

      Optional: False

  - **`baseline`** - A block that shows up to the top five error codes that occurred on the API calls made during the normal activity period. It also shows the average number of API calls made by the user agents during the normal activity period.

    Since: 1.07

    Optional: False

    - **`value`** - One of the top five error codes that occurred on the API calls made during the normal activity period, such as `AccessDeniedException`.

      Since: 1.07

      Optional: False

    - **`average`** - The historic average of API calls or errors per minute during the seven days preceding the Insights activity start time for the error code in the `value` field.

Since: 1.07

Optional: False

Example insightDetails block

The following is an example of an Insights event insightDetails block for an Insights event that occurred when the Application Auto Scaling API CompleteLifecycleAction was called an unusual number of times. For an example of a full Insights event, see CloudTrail log event reference (p. 400).

This example is from a starting Insights event, indicated by "state": "Start". The top user identities that called the APIs associated with the Insights event, CodeDeployRole1, CodeDeployRole2, and CodeDeployRole3, are shown in the attributions block, along with their average API call rates for this Insights event, and the baseline for the CodeDeployRole1 role. The attributions block also shows that the user agent is codedeploy.amazonaws.com, meaning the top user identities used the Amazon CodeDeploy console to run the API calls.

Because there are no error codes associated with the events that were analyzed to generate the Insights event (the value is null), the insight average for the error code is the same as the overall insight average for the entire Insights event, shown in the statistics block.

```
"insightDetails": {
  "state": "Start",
  "eventSource": "autoscaling.amazonaws.com",
  "eventName": "CompleteLifecycleAction",
  "insightType": "ApiCallRateInsight",
  "insightContext": {
    "statistics": {
      "baseline": {
        "average": 0.0000882145
      },
      "insight": {
        "average": 0.6
      },
      "insightDuration": 5,
      "baselineDuration": 11336
    },
    "attributions": [
      {
        "attribute": "userIdentityArn",
        "insight": [
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
            "average": 0.2
          },
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole2",
            "average": 0.2
          },
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole3",
            "average": 0.2
          }
        ],
        "baseline": [
          {
            "value": "arn:aws:sts::012345678901:assumed-role/CodeDeployRole1",
            "average": 0.0000882145
          }
        ]
```

```
            },
            {
              "attribute": "userAgent",
              "insight": [
                {
                  "value": "codedeploy.amazonaws.com",
                  "average": 0.6
                }
              ],
              "baseline": [
                {
                  "value": "codedeploy.amazonaws.com",
                  "average": 0.0000882145
                }
              ]
            },
            {
              "attribute": "errorCode",
              "insight": [
                {
                  "value": "null",
                  "average": 0.6
                }
              ],
              "baseline": [
                {
                  "value": "null",
                  "average": 0.0000882145
                }
              ]
            }
          ]
        }
      }
```

Non-API events captured by CloudTrail

In addition to logging Amazon API calls, CloudTrail captures other related events that might have a security or compliance impact on your Amazon account or that might help you troubleshoot operational problems.

Topics

- Amazon service events (p. 430)
- Amazon Web Services Management Console sign-in events (p. 431)

Amazon service events

CloudTrail supports logging non-API service events. These events are created by Amazon services but are not directly triggered by a request to a public Amazon API. For these events, the `eventType` field is `AwsServiceEvent`.

The following is an example scenario of an Amazon service event when a customer managed key is automatically rotated in Amazon Key Management Service (Amazon KMS). For more information about rotating KMS keys, see Rotating KMS keys.

```
{
    "eventVersion": "1.05",
```

```
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2019-06-02T00:06:08Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "234f004b-EXAMPLE",
    "readOnly": false,
    "resources": [
        {
            "ARN": "arn:aws:kms:us-east-2:123456789012:key/7944f0ec-EXAMPLE",
            "accountId": "123456789012",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
    "serviceEventDetails": {
        "keyId": "7944f0ec-EXAMPLE"
    }
}
```


Amazon Web Services Management Console sign-in events

Important
As of November 22, 2021, Amazon CloudTrail makes Amazon CloudFront events available only in the Region where the event was processed, China (Ningxia) Region, cn-northwest-1.
For trails monitoring global service events, be sure to convert *single-Region trails* in China (Beijing) Region, cn-north-1, to *multi-Region trails*, to include events from China (Ningxia) Region, cn-northwest-1. For more information about capturing CloudFront events, see Enabling and disabling global service event logging (p. 102) later in this section.

CloudTrail logs attempts to sign in to the Amazon Web Services Management Console, the Amazon Discussion Forums, and the Amazon Support Center. All IAM user and root user sign-in events, as well as all federated user sign-in events, generate records in CloudTrail log files. Amazon Web Services Management Console sign-in events are global service events. For information about getting and viewing logs, see Getting and viewing your CloudTrail log files (p. 144).

Topics

- Example records for IAM users (p. 431)
- Example event records for root users (p. 434)

Example records for IAM users

The following examples show event records for several IAM user sign-in scenarios.

Topics

- IAM user, successful sign-in without MFA (p. 432)
- IAM user, successful sign-in with MFA (p. 432)
- IAM user, unsuccessful sign-in (p. 433)

- IAM user, sign-in process checks for MFA (single MFA device type) (p. 433)
- IAM user, sign-in process checks for MFA (multiple MFA device types) (p. 434)

IAM user, successful sign-in without MFA

The following record shows that a user named Anaya successfully signed in to the Amazon Web Services Management Console without using multi-factor authentication (MFA).


```
{
   "Records":[
      {
         "eventVersion":"1.05",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/anaya",
            "accountId":"111122223333",
            "userName":"anaya"
         },
         "eventTime":"2022-11-10T16:24:34Z",
         "eventSource":"signin.amazonaws.com",
         "eventName":"ConsoleLogin",
         "awsRegion":"us-east-2",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:62.0) Gecko/20100101
 Firefox/62.0",
         "requestParameters":null,
         "responseElements":{
            "ConsoleLogin":"Success"
         },
         "additionalEventData":{
            "MobileVersion":"No",
            "LoginTo":"https://console.aws.amazon.com/sns",
            "MFAUsed":"No"
         },
         "eventID":"3fcfb182-98f8-4744-bd45-10a395ab61cb",
         "eventType": "AwsConsoleSignIn"
      }
   ]
}
```


IAM user, successful sign-in with MFA

The following record shows that an IAM user named Anaya successfully signed in to the Amazon Web Services Management Console using multi-factor authentication (MFA).


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/anaya",
        "accountId": "111122223333",
        "userName": "anaya"
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
```

```
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Success"
    },
    "additionalEventData": {
            "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
            "MobileVersion": "No",
            "MFAIdentifier": "arn:aws:iam::111122223333:u2f/user/anaya/default-
AAAAAAAABBBBBBBBCCCCCCCCDD",
            "MFAUsed": "Yes"
     },
    "eventID": "fed06f42-cb12-4764-8c69-121063dc79b9",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


IAM user, unsuccessful sign-in

The following record shows an unsuccessful sign-in attempt from an IAM user.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "accountId": "111122223333",
        "accessKeyId": "",
        "userName": "anaya"
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "errorMessage": "Failed authentication",
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Failure"
    },
    "additionalEventData": {
        "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
        "MobileVersion": "No",
        "MFAUsed": "Yes"
    },
    "eventID": "d38ce1b3-4575-4cb8-a632-611b8243bfc3",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


IAM user, sign-in process checks for MFA (single MFA device type)

The following shows that the sign-process checked whether multi-factor authentication (MFA) is required for an IAM user during sign-in. In this example, the `mfaType` value is `U2F MFA`, which indicates that the IAM user enabled either a single MFA device or multiple MFA devices of the same type (`U2F MFA`).


```
 {
    "eventVersion": "1.05",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "accountId": "111122223333",
        "accessKeyId": "",
        "userName": "anaya"
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "CheckMfa",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "requestParameters": null,
    "responseElements": {
        "CheckMfa": "Success"
    },
    "additionalEventData": {
        "MfaType": "U2F MFA"
    },
    "eventID": "f8ef8fc5-e3e8-4ee1-9d52-2f412ddf17e3",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


IAM user, sign-in process checks for MFA (multiple MFA device types)

The following shows that the sign-process checked whether multi-factor authentication (MFA) is required for an IAM user during sign-in. In this example, the `mfaType` value is `Multiple MFA Devices`, which indicates that the IAM user enabled multiple MFA device types.


```
 {
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "accountId": "111122223333",
        "accessKeyId": "",
        "userName": "anaya"
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "CheckMfa",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "requestParameters": null,
    "responseElements": {
        "CheckMfa": "Success"
    },
    "additionalEventData": {
        "MfaType": "Multiple MFA Devices"
    },
    "eventID": "f8ef8fc5-e3e8-4ee1-9d52-2f412ddf17e3",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


Example event records for root users

The following examples show event records for several `root` user sign-in scenarios.

Topics

- Root user, successful sign-in without MFA (p. 435)
- Root user, successful sign-in with MFA (p. 435)
- Root user, unsuccessful sign-in (p. 436)
- Root user, MFA changed (p. 436)
- Root user, password changed (p. 437)

Root user, successful sign-in without MFA

The following shows a successful sign-in event for a root user not using multi-factor authentication (MFA).


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": ""
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:62.0) Gecko/20100101
 Firefox/62.0",
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Success"
    },
    "additionalEventData": {
        "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
        "MobileVersion": "No",
        "MFAUsed": "No"
    },
    "eventID": "deb1e1f9-c99b-4612-8e9f-21f93b5d79c0",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


Root user, successful sign-in with MFA

The following shows a successful sign-in event for a root user using multi-factor authentication (MFA).


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": ""
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:62.0) Gecko/20100101
 Firefox/62.0",
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Success"
    },
    "additionalEventData": {
        "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
        "MobileVersion": "No",
        "MFAIdentifier": "arn:aws:iam::111122223333:mfa/root-account-mfa-device",
        "MFAUsed": "YES"
    },
    "eventID": "deb1e1f9-c99b-4612-8e9f-21f93b5d79c0",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


Root user, unsuccessful sign-in

The following shows an unsuccessful sign-in event for a root user not using MFA.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": ""
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36",
    "errorMessage": "Failed authentication",
    "requestParameters": null,
    "responseElements": {
        "ConsoleLogin": "Failure"
    },
    "additionalEventData": {
        "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
        "MobileVersion": "No",
        "MFAUsed": "No"
    },
    "eventID": "a4fbbe77-91a0-4238-804a-64314184edb6",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```


Root user, MFA changed

The following shows an example event for a root user changing multi-factor authentication (MFA) settings.


```
{
    "eventVersion": "1.05",
```

```
    "userIdentity": {
        "type": "Root",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-10-13T21:05:40Z"
            }
        }
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "EnableMFADevice",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Coral/Netty4",
    "requestParameters": {
        "userName": "AWS ROOT USER",
        "serialNumber": "arn:aws:iam::111122223333:mfa/root-account-mfa-device"
    },
    "responseElements": null,
    "requestID": "EXAMPLE4-2cf7-4a44-af00-f61f0EXAMPLE",
    "eventID": "EXAMPLEb-7dae-48cb-895f-20e86EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Root user, password changed

The following shows an example event for a root user changing their password.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:root",
        "accountId": "111122223333",
        "accessKeyId": ""
    },
    "eventTime": "2022-11-10T16:24:34Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "PasswordUpdated",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101
 Firefox/78.0",
    "requestParameters": null,
    "responseElements": {
        "PasswordUpdated": "Success"
    },
    "eventID": "EXAMPLEd-244c-4044-abbf-21c64EXAMPLE",
    "eventType": "AwsConsoleSignIn",
    "recipientAccountId": "111122223333"
}
```

Document history

The following table describes the important changes to the documentation for Amazon CloudTrail. For notification about updates to this documentation, you can subscribe to an RSS feed.

- **API version**: 2013-11-01
- **Latest documentation update**: 2023-08-01

Change	Description	Date
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Israel (Tel Aviv) Region. For more information, see CloudTrail supported Regions.	August 1, 2023
Added service support (p. 438)	This release supports Amazon HealthImaging. For more information, see CloudTrail supported services and integrations and Logging Amazon HealthImaging API calls using Amazon CloudTrail.	July 26, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon HealthImaging data stores by using advanced event selectors. For more information, see Logging data events.	July 26, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon Systems Manager control channels and Amazon Managed Blockchain networks by using advanced event selectors. For more information, see Logging data events.	June 21, 2023
Added functionality (p. 438)	You can now verify your CloudTrail Lake saved query results using the **aws cloudtrail verify-query-results** command. For more information, see Validate saved query results with the Amazon CLI.	June 21, 2023
Added service support (p. 438)	This release supports Amazon Verified Permissions. For more information, see CloudTrail supported services and integrations and Logging Amazon Verified Permissions API calls using Amazon CloudTrail.	June 13, 2023

Added functionality (p. 438)	You can now use CloudTrail Lake dashboards to visualize the events in an event data store. For more information, see View Lake dashboards.	June 13, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon Verified Permissions policy stores by using advanced event selectors. For more information, see Logging data events.	June 13, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on an Amazon CodeWhisperer profile by using advanced event selectors. For more information, see Logging data events.	June 6, 2023
Added functionality (p. 438)	You can now start and stop event ingestion on CloudTrail event data stores. For information about stopping event ingestion using the console, see Stop an event data store from ingesting events. For information about stopping event ingestion using the Amazon CLI, see Stop ingestion on an event data store.	June 2, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on an Amazon EMR write-ahead log workspace by using advanced event selectors. For more information, see Logging data events.	May 31, 2023
Added service support (p. 438)	This release supports Amazon Security Lake. For more information, see CloudTrail supported services and integrations and Logging Amazon Security Lake API calls using Amazon CloudTrail.	May 30, 2023
Updated documentation (p. 438)	Updated CloudTrail userIdentity element topic to include an example and field descriptions for a request made on behalf of an IAM Identity Center user. For more information, see CloudTrail userIdentity element.	May 23, 2023

Updated documentation (p. 438)	This update supports the following patch release for the CloudTrail Processing Library: aws-cloudtrail-processing-library-1.6.1.jar. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	May 23, 2023
Added functionality (p. 438)	CloudTrail Lake now supports all Presto functions and operators. For more information, see CloudTrail Lake SQL constraints.	May 9, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on an Amazon GuardDuty detector by using advanced event selectors. For more information, see Logging data events and Logging Amazon GuardDuty API calls with Amazon CloudTrail.	March 30, 2023
Updated documentation (p. 438)	Added new section about creating user-defined cost allocation tags for event data stores. For more information, see Creating user-defined cost allocation tags for CloudTrail Lake event data stores.	March 24, 2023
Added service support (p. 438)	This release supports Amazon Telco Network Builder (Amazon TNB). For more information, see CloudTrail supported services and integrations and Logging Amazon Telco Network Builder API calls using Amazon CloudTrail.	February 21, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon Cognito identity pools by using advanced event selectors. For more information, see Logging data events.	February 15, 2023
Updated documentation (p. 438)	Added new section about the learning resources available for CloudTrail Lake. For more information, see Learning resources.	February 9, 2023

Added functionality (p. 438)	You can now create CloudTrail Lake integrations with event sources outside of Amazon. You can log and store user activity data from any source in your hybrid environments, such as in-house or SaaS applications hosted on-premises or in the cloud, virtual machines, or containers. For more information, see Create an integration with an event source outside of Amazon.	January 31, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on CloudTrail `PutAuditEvents` activity on a CloudTrail Lake channel by using advanced event selectors. For more information, see Logging data events.	January 31, 2023
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Asia Pacific (Melbourne) Region. For more information, see CloudTrail supported Regions.	January 24, 2023
Updated documentation (p. 438)	Added new section about managing data consistency in CloudTrail, see Managing data consistency in CloudTrail.	January 18, 2023
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon SageMaker feature stores by using advanced event selectors. For more information, see Logging data events.	December 27, 2022
Added service support (p. 438)	This release supports Amazon Web Services Marketplace Discovery. See Amazon CloudTrail Supported Services and Integrations.	December 15, 2022
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon SageMaker metrics experiment trial components by using advanced event selectors. For more information, see Logging data events.	December 15, 2022

Added functionality (p. 438)	You can now create an event data store to include Amazon Config configuration items, and use the event data store to investigate non-compliant changes to your production environments. For more information, see Create an event data store for Amazon Config configuration items.	November 28, 2022
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Asia Pacific (Hyderabad) Region. For more information, see CloudTrail supported Regions.	November 22, 2022
Added functionality (p. 438)	You can now log CloudTrail data events on Amazon FinSpace environments by using advanced event selectors. For more information, see Logging data events.	November 18, 2022
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Europe (Spain) Region. For more information, see CloudTrail supported Regions.	November 16, 2022
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Europe (Zurich) Region. For more information, see CloudTrail supported Regions.	November 9, 2022
Added functionality (p. 438)	The management account for an Amazon Organizations organization can now add a delegated administrator to manage the organization's CloudTrail trails and event data stores. For more information, see Organization delegated administrator.	November 7, 2022
Added functionality (p. 438)	You can now enable Amazon Key Management Service encryption for a CloudTrail Lake event data store. For more information, see Create an event data store.	November 7, 2022

Added functionality (p. 438)	You can now save CloudTrail Lake query results to an Amazon S3 bucket when you run a query. For more information about running a query, see Run a query and save query results. For more information about downloading query results, see Get and download saved query results.	October 21, 2022
Added functionality (p. 438)	You can now copy CloudTrail trail events to a CloudTrail Lake event data store. For more information, see Copying trail events to CloudTrail Lake.	September 19, 2022
Updated documentation (p. 438)	Added list of supported Amazon CloudWatch metrics for CloudTrail Lake. For more information, see Supported CloudWatch metrics.	September 16, 2022
Added functionality (p. 438)	You can now view CloudTrail service-linked channels using the Amazon CLI. For more information, see Viewing service-linked channels for CloudTrail by using the Amazon CLI.	September 9, 2022
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Middle East (UAE) Region. For more information, see CloudTrail supported Regions.	August 30, 2022
Updated documentation (p. 438)	Added list of services that report TLS details to CloudTrail. For more information, see Services that support TLS details in CloudTrail.	July 19, 2022
Changed functionality (p. 438)	CloudTrail has changed the name of the managed policy `AWSCloudTrailReadOnlyAccess` to `AWSCloudTrail_ReadOnlyAccess`. Permissions in this policy have been scoped down. By default, the policy no longer grants permission to list all Amazon S3 buckets, Amazon Lambda functions, or Amazon KMS aliases. For more information, see Read-only access.	June 6, 2022

Changed functionality (p. 438)	As a security best practice, you can now add an `aws:SourceArn` or `aws:SourceAccount` condition key to an `s3:GetBucketAcl` ACL checking block in Amazon S3 bucket policies. For more information, see Configure Amazon S3 bucket policies for CloudTrail.	May 11, 2022
Changed functionality (p. 438)	Starting Feb 24, 2022, Amazon CloudTrail began changing the `userAgent` and `sourceIPAddress` field values in any event that originated from an Amazon Web Services Management Console session where a proxy client was used. For these events, CloudTrail replaces the values of the `userAgent` and `sourceIPAddress` fields with `AWS Internal`. CloudTrail made this change to standardize how it logs information for service actions across all Amazon services. For more information, see CloudTrail record contents.	April 12, 2022
Added service support (p. 438)	This release supports Amazon GameSparks. See Amazon CloudTrail Supported Services and Integrations.	March 24, 2022
Added service support (p. 438)	This release supports Amazon App Mesh Envoy Management Service. See Amazon CloudTrail Supported Services and Integrations.	March 18, 2022
Updated documentation (p. 438)	New query examples have been added for CloudTrail Lake, a new feature that lets you run fine-grained, multiple-field SQL queries on your events. Also, a new field, `BytesScanned`, has been added to the query metadata results of `DescribeQuery` and `GetQueryResults` operations. For more information, see Working with CloudTrail Lake.	March 4, 2022

Changed functionality (p. 438)	CloudTrail now removes the account ID of the Amazon S3 bucket owner in the `resources` block of a data event if both of the following conditions are met: the data event API call is from a different Amazon account than the Amazon S3 bucket owner, and the API caller received an `AccessDenied` error that was only for the caller account. For more information, see Redacting bucket owner account IDs for data events called by other accounts.	March 3, 2022
Updated documentation (p. 438)	This update supports the following release for the CloudTrail Processing Library: Added support for implementing a custom S3 manager, event logging to log file parsing-related exceptions, support for parsing an optional `errorCode` field in `insightDetails`, and updated the account ID parsing regex to accept non-numerical values. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	January 28, 2022
Added functionality (p. 438)	CloudTrail introduces CloudTrail Lake, a new feature that lets you run fine-grained, multiple-field SQL queries on your events. Events are aggregated into event data stores, which are immutable collections of events based on criteria that you select by applying advanced event selectors. For more information, see Working with CloudTrail Lake.	January 5, 2022
New Region support (p. 438)	CloudTrail expanded support to a new Region, the Asia Pacific (Jakarta) Region. For more information, see CloudTrail supported Regions.	December 13, 2021
Added service support (p. 438)	This release supports Amazon WorkSpaces Web. See Amazon CloudTrail Supported Services and Integrations.	December 3, 2021

Added functionality (p. 438)	You can now log CloudTrail data events on Amazon Glue tables created by Lake Formation by using advanced event selectors. For more information, see Logging data events.	November 30, 2021
Changed functionality (p. 438)	As a security best practice, you can now add an `aws:SourceArn` or `aws:SourceAccount` condition key to Amazon KMS key policies and Amazon S3 bucket policies. For more information, see Configure Amazon KMS key policies for CloudTrail and Configure Amazon S3 bucket policies for CloudTrail.	November 15, 2021
Added service support (p. 438)	This release supports Amazon Resilience Hub. See Amazon CloudTrail Supported Services and Integrations.	November 10, 2021
Added functionality (p. 438)	A new CloudTrail Insights event type is available: error rate Insights events. An error rate Insights event captures unusual activity on an error that occurs on APIs called in your account. For more information, see Logging Insights events for trails.	November 10, 2021
Added functionality (p. 438)	You can now log CloudTrail data events on DynamoDB streams by using advanced event selectors. For more information, see Logging data events.	September 22, 2021
Added functionality (p. 438)	You can now log data events on Amazon S3 access points. You can log Amazon S3 access point data events by using advanced event selectors. For more information, see Logging data events.	August 24, 2021

Changed functionality (p. 438)	When you configure a trail to send notifications to Amazon SNS, CloudTrail adds a policy statement to your SNS topic access policy that allows CloudTrail to send content to an SNS topic. As a security best practice, we recommend adding an `aws:SourceArn` or `aws:SourceAccount` condition key to the CloudTrail policy statement. For more information, see Amazon SNS topic policy for CloudTrail.	August 16, 2021
Added service support (p. 438)	This release supports Amazon Route 53 Application Recovery Controller. See Amazon CloudTrail Supported Services and Integrations.	July 27, 2021
Added functionality (p. 438)	You can now log data events on Amazon EBS direct APIs run on EBS snapshots. You can log Amazon EBS direct API data events by using advanced event selectors. For more information, see Logging data events.	July 27, 2021
Changed functionality (p. 438)	When CloudTrail processes data events, it preserves numbers in their original format, whether that is an integer (`int`) or a `float`. In events that have integers in the fields of a data event, CloudTrail historically processed these numbers as floats. Now, CloudTrail keeps the original format of integers in data events. For more information, see Using the CloudTrail Processing Library.	July 13, 2021
Added service support (p. 438)	This release supports Amazon BugBust. See Amazon CloudTrail Supported Services and Integrations.	June 24, 2021
Added service support (p. 438)	This release supports Amazon Managed Grafana and Amazon Managed Service for Prometheus. See Amazon CloudTrail Supported Services and Integrations.	June 2, 2021

Added service support (p. 438)	This release supports Amazon App Runner. See Amazon CloudTrail Supported Services and Integrations.	May 18, 2021
Added service support (p. 438)	This release supports Amazon Systems Manager Incident Manager. See Amazon CloudTrail Supported Services and Integrations.	May 10, 2021
Updated documentation (p. 438)	This update describes data event logging requirements for Amazon Config conformance packs, especially for compliance frameworks such as HIPAA or FedRAMP. For more information, see Logging data events.	May 7, 2021
Added service support (p. 438)	This release supports Service Quotas and Amazon EBS direct APIs. See Amazon CloudTrail Supported Services and Integrations.	April 13, 2021
Added functionality (p. 438)	After an IAM administrator configures Amazon STS, CloudTrail logs `sourceIdentity` information in events when users assume an IAM role, or perform any actions with the assumed role. For more information, see CloudTrail userIdentity Element.	April 13, 2021
Updated documentation (p. 438)	This update documents limits, in kilobytes (KB), for content in some CloudTrail event record fields. For more information, see CloudTrail record contents.	April 8, 2021
Added functionality (p. 438)	After an IAM administrator configures Amazon STS, CloudTrail logs `sourceIdentity` information in events when users assume an IAM role, or perform any actions with the assumed role. For more information, see CloudTrail userIdentity Element.	April 6, 2021
Added functionality (p. 438)	You can now log data events on Amazon DynamoDB tables. You can log DynamoDB data events by using either event selectors or advanced event selectors. For more information, see Logging data events.	March 23, 2021

Added service support (p. 438)	This release supports Amazon Managed Workflows for Apache Airflow. See Amazon CloudTrail Supported Services and Integrations.	March 22, 2021
Added functionality (p. 438)	You can now log data events on S3 Object Lambda access points if you have opted in to use advanced event selectors. For more information, see Logging data events.	March 18, 2021
Added service support (p. 438)	This release supports Amazon Fault Injection Simulator. See Amazon CloudTrail Supported Services and Integrations.	March 15, 2021
Added functionality (p. 438)	You can now log data events on Ethereum nodes in Amazon Managed Blockchain if you have opted in to use advanced event selectors. For more information, see Logging data events.	March 1, 2021
Added service support (p. 438)	This release supports Amazon Managed Blockchain and the preview of Ethereum for Managed Blockchain. See Amazon CloudTrail Supported Services and Integrations.	February 4, 2021
Added service support (p. 438)	This release supports Amazon Amplify. See Amazon CloudTrail Supported Services and Integrations.	February 3, 2021
Added service support (p. 438)	This release supports Amazon Lookout for Metrics. See Amazon CloudTrail Supported Services and Integrations.	February 1, 2021
Updated documentation (p. 438)	This update supports the following patch release for the CloudTrail Processing Library: Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.4.0.jar. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	January 12, 2021
Added functionality (p. 438)	You can now log data events on Amazon S3 on Amazon Outposts. For more information, see Logging data events.	December 21, 2020

Added service support (p. 438)	This release supports Amazon Lookout for Equipment, Amazon Well-Architected Tool, and Amazon Location Service. See Amazon CloudTrail Supported Services and Integrations.	December 16, 2020
Added service support (p. 438)	This release supports Amazon IoT Greengrass V2. See Amazon CloudTrail Supported Services and Integrations.	December 15, 2020
Added service support (p. 438)	This release supports Amazon EMR on EKS. See Amazon CloudTrail Supported Services and Integrations.	December 10, 2020
Added service support (p. 438)	This release supports Amazon Audit Manager and Amazon HealthLake. See Amazon CloudTrail Supported Services and Integrations.	December 8, 2020
Added service support (p. 438)	This release supports Amazon Lookout for Vision. See Amazon CloudTrail Supported Services and Integrations.	December 1, 2020
Added functionality (p. 438)	The Amazon CloudTrail event version is now 1.08. Version 1.08 introduces new fields for CloudTrail. For more information, see CloudTrail record contents.	November 24, 2020
Added functionality (p. 438)	Amazon CloudTrail introduces advanced event selectors for data events. Advanced event selectors allow finer-grained control over the data events that you log to your trail. You can include or exclude data events for specific Amazon resources, and select specific APIs on those resources to log to your trail. For more information, see Logging data events.	November 24, 2020
Added service support (p. 438)	This release supports Amazon Network Firewall. See Amazon CloudTrail Supported Services and Integrations.	November 17, 2020
Added service support (p. 438)	This release supports Amazon Trusted Advisor. See Amazon CloudTrail Supported Services and Integrations.	October 22, 2020

Updated documentation (p. 438)	Added two new examples of event records for root user sign-in events. For more information, see Amazon Console sign-in events.	October 13, 2020
Changed functionality (p. 438)	Permissions in the `AWSCloudTrail_FullAccess` policy have been narrowed. This policy no longer allows you to delete Amazon SNS topics or Amazon S3 buckets, and the `getObject` action has been removed. For more information, see Granting custom permissions for CloudTrail users.	September 29, 2020
Updated documentation (p. 438)	This update supports the following patch release for the CloudTrail Processing Library: Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.3.0.jar. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	August 28, 2020
Added service support (p. 438)	This release supports Amazon Outposts. See Amazon CloudTrail Supported Services and Integrations.	August 28, 2020
Added functionality (p. 438)	Amazon CloudTrail Insights introduces attribution fields for CloudTrail Insights events. Attribution fields show the top user identities, user agents, and error codes that are associated with the anomalous activity that triggers Insights events. For comparison, attribution fields also show the top user identities, user agents, and error codes associated with normal, or baseline, activity. For more information, see Logging Insights events for trails.	August 13, 2020

Added functionality (p. 438)	The Amazon CloudTrail console has a new look that's designed to make it easier to use. The Amazon CloudTrail User Guide has been updated with changes to procedures for how to perform tasks in the console, such as creating trails, updating trails, and downloading event history.	August 13, 2020
Added service support (p. 438)	This release supports Amazon Interactive Video Service. See Amazon CloudTrail Supported Services and Integrations.	July 15, 2020
Added service support (p. 438)	This release supports Amazon Honeycode. See Amazon CloudTrail Supported Services and Integrations.	June 24, 2020
Added service support (p. 438)	This release supports Amazon Macie. See Amazon CloudTrail Supported Services and Integrations.	May 19, 2020
Added service support (p. 438)	This release supports Amazon Kendra. See Amazon CloudTrail Supported Services and Integrations.	May 13, 2020
Added service support (p. 438)	This release supports Amazon IoT SiteWise. See Amazon CloudTrail Supported Services and Integrations.	April 29, 2020
Added Region support (p. 438)	This release supports an additional Region: Europe (Milan). See Amazon CloudTrail Supported Regions.	April 28, 2020
Added service and Region support (p. 438)	This release supports Amazon AppFlow. See Amazon CloudTrail Supported Services and Integrations. Support has also been added for the Africa (Cape Town) Region. See Amazon CloudTrail Supported Regions.	April 22, 2020

Added functionality (p. 438)	High-volume Amazon KMS actions such as `Encrypt`, `Decrypt`, and `GenerateDataKey` are now logged as **Read** events. If you choose to log all Amazon KMS events on your trail, and also choose to log **Write** management events, your trail logs relevant Amazon KMS actions like `Disable`, `Delete` and `ScheduleKey`.	April 7, 2020
Added service support (p. 438)	This release supports Amazon CodeGuru Reviewer. See Amazon CloudTrail Supported Services and Integrations.	February 7, 2020
Added service support (p. 438)	This release supports Amazon Managed Apache Cassandra Service. See Amazon CloudTrail Supported Services and Integrations.	January 17, 2020
Added service support (p. 438)	This release supports Amazon Connect. See Amazon CloudTrail Supported Services and Integrations.	December 13, 2019
Updated documentation (p. 438)	This update supports the following patch release for the CloudTrail Processing Library: Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.2.0.jar. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	November 21, 2019
Added functionality (p. 438)	This release supports Amazon CloudTrail Insights for helping you detect unusual activity in your account. See Logging Insights events for Trails.	November 20, 2019
Added functionality (p. 438)	This release adds an option for filtering Amazon Key Management Service events out of a trail. See Creating a Trail.	November 20, 2019
Added service support (p. 438)	This release supports AWS CodeStar Notifications. See Amazon CloudTrail Supported Services and Integrations.	November 7, 2019

Added functionality (p. 438)	This release supports adding tags when you create a trail in CloudTrail, whether you use the CloudTrail console or API. This release adds two new APIs, `GetTrail` and `ListTrails`.	November 1, 2019
Added service support (p. 438)	This release supports Amazon App Mesh. See Amazon CloudTrail Supported Services and Integrations.	October 17, 2019
Added service support (p. 438)	This release supports Amazon Translate. See Amazon CloudTrail Supported Services and Integrations.	October 17, 2019
Documentation update (p. 438)	The Unsupported Services topic has been restored and updated to include only those Amazon services that do not currently log events in CloudTrail. See CloudTrail Unsupported Services.	October 7, 2019
Documentation update (p. 438)	The documentation has been updated with changes to the `AWSCloudTrailFullAccess` policy. A policy example that shows equivalent permissions to `AWSCloudTrailFullAccess` has been updated to restrict the resources on which the `iam:PassRole` action can act to those matching the following condition statement: `"iam:PassedToService": "cloudtrail.amazonaws.com"`. See Amazon CloudTrail Identity-Based Policy Examples.	September 24, 2019
Documentation update (p. 438)	The documentation has been updated with a new topic, Managing CloudTrail Costs, to help you get the log data you need out of CloudTrail while staying within a budget.	September 3, 2019
Added service support (p. 438)	This release supports Amazon Control Tower. See Amazon CloudTrail Supported Services and Integrations.	August 13, 2019
Added Region support (p. 438)	This release supports an additional Region: Middle East (Bahrain). See Amazon CloudTrail Supported Regions.	July 29, 2019

Documentation update (p. 438)	The documentation has been updated with information about security for CloudTrail. See Security in Amazon CloudTrail.	July 3, 2019
Added service support (p. 438)	This release supports Amazon Ground Station. See Amazon CloudTrail Supported Services and Integrations.	June 6, 2019
Added service support (p. 438)	This release supports Amazon IoT Things Graph. See Amazon CloudTrail Supported Services and Integrations.	June 4, 2019
Added service support (p. 438)	This release supports Amazon AppStream 2.0. See Amazon CloudTrail Supported Services and Integrations.	April 25, 2019
Added Region support (p. 438)	This release supports an additional Region: Asia Pacific (Hong Kong). See Amazon CloudTrail Supported Regions.	April 24, 2019
Added service support (p. 438)	This release supports Amazon Kinesis Data Analytics. See Amazon CloudTrail Supported Services and Integrations.	March 22, 2019
Added service support (p. 438)	This release supports Amazon Backup. See Amazon CloudTrail Supported Services and Integrations.	February 4, 2019
Added service support (p. 438)	This release supports Amazon WorkLink. See Amazon CloudTrail Supported Services and Integrations.	January 23, 2019
Added service support (p. 438)	This release supports Amazon Cloud9. See Amazon CloudTrail Supported Services and Integrations.	January 21, 2019
Added service support (p. 438)	This release supports AWS Elemental MediaLive. See Amazon CloudTrail Supported Services and Integrations.	January 19, 2019
Added service support (p. 438)	This release supports Amazon Comprehend. See Amazon CloudTrail Supported Services and Integrations.	January 18, 2019
Added service support (p. 438)	This release supports AWS Elemental MediaPackage. See Amazon CloudTrail Supported Services and Integrations.	December 21, 2018

Added Region support (p. 438)	This release supports an additional Region: EU (Stockholm). See Amazon CloudTrail Supported Regions.	December 11, 2018
Documentation update (p. 438)	The documentation has been updated with information about supported and unsupported services. See Amazon CloudTrail Supported Services and Integrations.	December 3, 2018
Added service support (p. 438)	This release supports Amazon Resource Access Manager (Amazon RAM). See Amazon CloudTrail Supported Services and Integrations.	November 20, 2018
Updated functionality (p. 438)	This release supports creating a trail in CloudTrail that logs events for all Amazon accounts in an organization in Amazon Organizations. See Creating a Trail for an Organization.	November 19, 2018
Added service support (p. 438)	This release supports Amazon Pinpoint SMS and Voice API. See Amazon CloudTrail Supported Services and Integrations.	November 16, 2018
Added service support (p. 438)	This release supports Amazon IoT Greengrass. See Amazon CloudTrail Supported Services and Integrations.	October 29, 2018
Updated documentation (p. 438)	This update supports the following patch release for the CloudTrail Processing Library: Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.1.3.jar. For more information, see Using the CloudTrail Processing Library and the CloudTrail Processing Library on GitHub.	October 18, 2018
Added functionality (p. 438)	This release supports using additional filters in **Event history**. See Viewing CloudTrail Events in the CloudTrail Console.	October 18, 2018

Added functionality (p. 438)	This release supports using Amazon Virtual Private Cloud (Amazon VPC) to establish a private connection between your VPC and Amazon CloudTrail. See Using Amazon CloudTrail with Interface VPC Endpoints.	August 9, 2018
Added service support (p. 438)	This release supports Amazon Data Lifecycle Manager. See Amazon CloudTrail Supported Services and Integrations.	July 24, 2018
Added service support (p. 438)	This release supports Amazon MQ. See Amazon CloudTrail Supported Services and Integrations.	July 19, 2018
Added service support (p. 438)	This release supports Amazon Mobile CLI. See Amazon CloudTrail Supported Services and Integrations.	June 29, 2018
Amazon CloudTrail documentation history notification available through RSS feed (p. 438)	You can now receive notification about updates to the Amazon CloudTrail documentation by subscribing to an RSS feed.	June 29, 2018

Earlier updates

The following table describes the documentation release history of Amazon CloudTrail prior to June 29, 2018.

Change	Description	Release Date
Added service support	This release supports Amazon RDS Performance Insights. For more information, see CloudTrail Supported Services and Integrations.	June 21, 2018
Added functionality	This release supports logging all CloudTrail management events in Event history. For more information, see Viewing events with CloudTrail Event history (p. 50).	June 14, 2018
Added service support	This release supports Amazon Billing and Cost Management. See CloudTrail supported services and integrations (p. 24).	June 7, 2018
Added service support	This release supports Amazon Elastic Container Service for Kubernetes (Amazon EKS). See CloudTrail supported services and integrations (p. 24).	June 5, 2018
Updated documentation	This update supports the following patch release for the CloudTrail Processing Library:	May 16, 2018

Change	Description	Release Date
	• Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.1.2.jar. For more information, see Using the CloudTrail Processing Library (p. 327) and the CloudTrail Processing Library on GitHub.	
Added service support	This release supports Amazon Billing and Cost Management. See CloudTrail supported services and integrations (p. 24).	June 7, 2018
Added service support	This release supports Amazon Elastic Container Service for Kubernetes (Amazon EKS). See CloudTrail supported services and integrations (p. 24).	June 5, 2018
Updated documentation	This update supports the following patch release for the CloudTrail Processing Library: • Update the .jar file references in the user guide to use the latest version, aws-cloudtrail-processing-library-1.1.2.jar. For more information, see Using the CloudTrail Processing Library (p. 327) and the CloudTrail Processing Library on GitHub.	May 16, 2018
Added service support	This release supports Amazon X-Ray. See CloudTrail supported services and integrations (p. 24).	April 25, 2018
Added service support	This release supports Amazon IoT Analytics. See CloudTrail supported services and integrations (p. 24).	April 23, 2018
Added service support	This release supports Secrets Manager. See CloudTrail supported services and integrations (p. 24).	April 10, 2018
Added service support	This release supports Amazon Rekognition. See CloudTrail supported services and integrations (p. 24).	April 6, 2018
Added service support	This release supports Amazon Private Certificate Authority (PCA). See CloudTrail supported services and integrations (p. 24).	April 4, 2018
Added functionality	This release supports making it easier to search CloudTrail log files with Amazon Athena. You can automatically create tables for querying logs directly from the CloudTrail console, and use those tables to run queries in Athena. For more information, see CloudTrail supported services and integrations (p. 24) and Creating a Table for CloudTrail Logs in the CloudTrail Console.	March 15, 2018
Added service support	This release supports Amazon AppSync. See CloudTrail supported services and integrations (p. 24).	February 13, 2018

Change	Description	Release Date
Added Region support	This release supports an additional Region: Asia Pacific (Osaka) (ap-northeast-3). See CloudTrail supported Regions (p. 15).	February 12, 2018
Added service support	This release supports Amazon Shield. See CloudTrail supported services and integrations (p. 24).	February 12, 2018
Added service support	This release supports Amazon SageMaker. See CloudTrail supported services and integrations (p. 24).	January 11, 2018
Added service support	This release supports Amazon Batch. See CloudTrail supported services and integrations (p. 24).	January 10, 2018
Added functionality	This release supports extending the amount of account activity that is available in CloudTrail event history to 90 days. You can also customize the display of columns to improve the view of your CloudTrail events. For more information, see Viewing events with CloudTrail Event history (p. 50).	December 12, 2017
Added service support	This release supports Amazon WorkMail. See CloudTrail supported services and integrations (p. 24).	December 12, 2017
Added service support	This release supports Alexa for Business, AWS Elemental MediaConvert, and AWS Elemental MediaStore. See CloudTrail supported services and integrations (p. 24).	December 1, 2017
Added functionality and documentation	This release supports logging data events for Amazon Lambda functions. For more information, see Logging data events (p. 242).	November 30, 2017
Added functionality and documentation	This release supports logging data events for Amazon Lambda functions. For more information, see Logging data events (p. 242).	November 30, 2017
Added functionality and documentation	This release supports the following updates to the CloudTrail Processing Library: • Add support for Boolean identification of management events. • Update the CloudTrail event version to 1.06. For more information, see Using the CloudTrail Processing Library (p. 327) and the CloudTrail Processing Library on GitHub.	November 30, 2017
Added service support	This release supports Amazon Glue. See CloudTrail supported services and integrations (p. 24).	November 7, 2017
New documentation	This release adds a new topic, Quotas in Amazon CloudTrail (p. 40).	October 19, 2017

Change	Description	Release Date
Updated documentation	This release updates the documentation of APIs supported in CloudTrail event history for Amazon Athena, Amazon CodeBuild, Amazon Elastic Container Registry, and Amazon Migration Hub.	October 13, 2017
Added service support	This release supports Amazon Chime. See CloudTrail supported services and integrations (p. 24).	September 27, 2017
Added functionality and documentation	This release supports configuring data event logging in the China (Beijing) Region for all Amazon S3 buckets in your Amazon account. See Logging data events (p. 242).	September 27, 2017
Added functionality and documentation	This release supports configuring data event logging for all Amazon S3 buckets in your Amazon account. See Logging data events (p. 242).	September 20, 2017
Added service support	This release supports Amazon Lex. See CloudTrail supported services and integrations (p. 24).	August 15, 2017
Added service support	This release supports Amazon Migration Hub. See CloudTrail supported services and integrations (p. 24).	August 14, 2017
Added functionality and documentation	This release supports CloudTrail being enabled by default for all Amazon accounts. The past seven days of account activity are available in CloudTrail event history, and the most recent events appear on the console dashboard. The feature formerly known as **API activity history** has been replaced by **Event history**. For more information, see How CloudTrail works (p. 1).	August 14, 2017
Added functionality and documentation	This release supports downloading events from the CloudTrail console on the API activity history page. You can download events in JSON or CSV format. For more information, see Downloading events (p. 54).	July 27, 2017
Added functionality	This release supports logging Amazon S3 object level API operations in two additional Regions, Europe (London) and Canada (Central). For more information, see Working with CloudTrail log files (p. 236).	July 19, 2017
Added service support	This release supports looking up APIs for Amazon CloudWatch Events in the CloudTrail API activity history feature.	June 27, 2017

Change	Description	Release Date
Added functionality and documentation	This release supports additional APIs in the CloudTrail API activity history feature for the following services: • Amazon CloudHSM • Amazon Cognito • Amazon DynamoDB • Amazon EC2 • Kinesis • Amazon Storage Gateway	June 27, 2017
Added service support	This release supports AWS CodeStar. See CloudTrail supported services and integrations (p. 24).	June 14, 2017
Added functionality and documentation	This release supports the following updates to the CloudTrail Processing Library: • Add support for different formats for SQS messages from the same SQS queue to identify CloudTrail log files. The following formats are supported: • Notifications that CloudTrail sends to an SNS topic • Notifications that Amazon S3 sends to an SNS topic • Notifications that Amazon S3 sends directly to an SQS queue • Add support for the `deleteMessageUponFailure` property, which you can use to delete messages that can't be processed. For more information, see Using the CloudTrail Processing Library (p. 327) and the CloudTrail Processing Library on GitHub.	June 1, 2017
Added service support	This release supports Amazon Athena. See CloudTrail supported services and integrations (p. 24).	May 19, 2017
Added functionality	This release supports sending data events to Amazon CloudWatch Logs. For more information about configuring your trail to log data events, see Data events (p. 243). For more information about sending events to CloudWatch Logs, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).	May 9, 2017
Added service support	This release supports the Amazon Web Services Marketplace Metering Service. See CloudTrail supported services and integrations (p. 24).	May 2, 2017
Added service support	This release supports Amazon QuickSight. See CloudTrail supported services and integrations (p. 24).	April 28, 2017

Change	Description	Release Date
Added functionality and documentation	This release supports an updated console experience for creating new trails. You can now configure a new trail to log management and data events. For more information, see Creating a trail (p. 78).	April 11, 2017
Added documentation	If CloudTrail is not delivering logs to your S3 bucket or sending SNS notifications from some Regions in your account, you may need to update the policies. To learn more about updating your S3 bucket policy, see Common Amazon S3 policy configuration errors (p. 366). To learn more about updating your SNS topic policy, see Common SNS policy configuration errors (p. 373).	March 31, 2017
Added service support	This release supports Amazon Organizations. See CloudTrail supported services and integrations (p. 24).	February 27, 2017
Added functionality and documentation	This release supports an updated console experience for configuring trails for logging management and data events. For more information, see Working with CloudTrail log files (p. 236).	February 10, 2017
Added service support	This release supports Amazon Cloud Directory. See CloudTrail supported services and integrations (p. 24).	January 26, 2017
Added functionality and documentation	This release supports looking up APIs for Amazon CodeCommit, Amazon GameLift, and Amazon Managed Services in the CloudTrail API activity history.	January 26, 2017
Added functionality	This release supports integration with the Amazon Health Dashboard. You can use the Amazon Health Dashboard to identify if your trails are unable to deliver logs to an SNS topic or S3 bucket. This can occur when there is an issue with the policy for the S3 bucket or SNS topic. Amazon Health Dashboard notifies you about the affected trails and recommends ways to fix the policy. For more information, see the Amazon Health User Guide.	January 24, 2017
Added functionality and documentation	This release supports filtering by event source in the CloudTrail console. Event source shows the Amazon service to which the request was made. For more information, see Viewing CloudTrail events in the CloudTrail console (p. 51).	January 12, 2017
Added service support	This release supports Amazon CodeCommit. See CloudTrail supported services and integrations (p. 24).	January 11, 2017
Added service support	This release supports Amazon Lightsail. See CloudTrail supported services and integrations (p. 24).	December 23, 2016

Change	Description	Release Date
Added service support	This release supports Amazon Managed Services. See CloudTrail supported services and integrations (p. 24).	December 21, 2016
Added Region support	This release supports the Europe (London) Region. See CloudTrail supported Regions (p. 15).	December 13, 2016
Added Region support	This release supports the Canada (Central) Region. See CloudTrail supported Regions (p. 15).	December 8, 2016
Added service support	This release supports Amazon CodeBuild See CloudTrail supported services and integrations (p. 24). This release supports Amazon Health. See CloudTrail supported services and integrations (p. 24). This release supports Amazon Step Functions. See CloudTrail supported services and integrations (p. 24).	December 1, 2016
Added service support	This release supports Amazon Polly. See CloudTrail supported services and integrations (p. 24).	November 30, 2016
Added service support	This release supports Amazon OpsWorks for Chef Automate. See CloudTrail supported services and integrations (p. 24).	November 23, 2016
Added functionality and documentation	This release supports configuring your trail to log read-only, write-only, or all events. CloudTrail supports logging Amazon S3 object level API operations such as `GetObject`, `PutObject`, and `DeleteObject`. You can configure your trails to log object level API operations. For more information, see Working with CloudTrail log files (p. 236).	November 21, 2016
Added functionality and documentation	This release supports additional values for the `type` field in the `userIdentity` element: `AWSAccount` and `AWSService`. For more information, see the Fields (p. 417) for `userIdentity`.	November 16, 2016
Added service support	This release supports Application Auto Scaling. See CloudTrail supported services and integrations (p. 24).	October 31, 2016
Added Region support	This release supports the US East (Ohio) Region. See CloudTrail supported Regions (p. 15).	October 17, 2016
Added functionality and documentation	This release supports logging non-API Amazon service events. For more information, see Amazon service events (p. 430).	September 23, 2016
Added functionality and documentation	This release supports using the CloudTrail console to view resource types that are supported by Amazon Config. For more information, see Viewing resources referenced with Amazon Config (p. 55).	July 7, 2016

Change	Description	Release Date
Added service support	This release supports Amazon Service Catalog. See CloudTrail supported services and integrations (p. 24).	July 6, 2016
Added functionality and documentation	This release supports CloudWatch Logs in the China (Beijing) Region. For more information about CloudTrail support for CloudWatch Logs monitoring, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).	June 30, 2016
Added service support	This release supports Amazon Elastic File System (Amazon EFS). See CloudTrail supported services and integrations (p. 24).	June 28, 2016
Added Region support	This release supports one additional Region: ap-south-1 (Asia Pacific (Mumbai)). See CloudTrail supported Regions (p. 15).	June 27, 2016
Added service support	This release supports Amazon Application Discovery Service. See CloudTrail supported services and integrations (p. 24).	May 12, 2016
Added service support	This release supports CloudWatch Logs in the South America (São Paulo) Region. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).	May 6, 2016
Added service support	This release supports Amazon WAF. See CloudTrail supported services and integrations (p. 24).	April 28, 2016
Added service support	This release supports Amazon Web Services Support. See CloudTrail supported services and integrations (p. 24).	April 21, 2016
Added service support	This release supports Amazon Inspector. See CloudTrail supported services and integrations (p. 24).	April 20, 2016
Added service support	This release supports Amazon IoT. See CloudTrail supported services and integrations (p. 24).	April 11, 2016
Added functionality and documentation	This release supports logging Amazon Security Token Service (Amazon STS) API calls made with Security Assertion Markup Language (SAML) and web identity federation. For more information, see Values for Amazon STS APIs with SAML and web identity federation (p. 421).	March 28, 2016
Added service support	This release supports Amazon Certificate Manager. See CloudTrail supported services and integrations (p. 24).	March 25, 2016
Added service support	This release supports Amazon Kinesis Data Firehose. See CloudTrail supported services and integrations (p. 24).	March 17, 2016
Added service support	This release supports Amazon CloudWatch Logs. See CloudTrail supported services and integrations (p. 24).	March 10, 2016

Change	Description	Release Date
Added functionality and documentation	This release supports turning on CloudTrail across all Regions and support for multiple trails per Region. For more information, see Receiving CloudTrail log files from multiple Regions (p. 275) and Create multiple trails (p. 237).	February 18, 2016
Added service support	This release supports Amazon Cognito. See CloudTrail supported services and integrations (p. 24).	February 18, 2016
Added service support	This release supports Amazon Database Migration Service. See CloudTrail supported services and integrations (p. 24).	February 4, 2016
Added service support	This release supports Amazon GameLift (Amazon GameLift). See CloudTrail supported services and integrations (p. 24).	January 27, 2016
Added service support	This release supports Amazon CloudWatch Events. See CloudTrail supported services and integrations (p. 24).	January 16, 2016
Added Region support	This release supports one additional Region: ap-northeast-2 (Asia Pacific (Seoul)). See CloudTrail supported Regions (p. 15).	January 6, 2016
Added service support	This release supports Amazon Elastic Container Registry (Amazon ECR). See CloudTrail supported services and integrations (p. 24).	December 21, 2015
Added service support	This release supports Amazon Machine Learning. See CloudTrail supported services and integrations (p. 24).	December 10, 2015
Added functionality and documentation	This release supports log file encryption, log file integrity validation, and tagging. For more information, see Encrypting CloudTrail log files with Amazon KMS keys (SSE-KMS) (p. 386), Validating CloudTrail log file integrity (p. 307), and Updating a trail (p. 89).	October 1, 2015
Added service support	This release supports Amazon OpenSearch Service. See CloudTrail supported services and integrations (p. 24).	October 1, 2015
Added service support	This release supports Amazon S3 bucket level events. See CloudTrail supported services and integrations (p. 24).	September 1, 2015
Added service support	This release supports Amazon Device Farm. See CloudTrail supported services and integrations (p. 24).	July 13, 2015
Added service support	This release supports Amazon API Gateway. See CloudTrail supported services and integrations (p. 24).	July 9, 2015
Added service support	This release supports CodePipeline. See CloudTrail supported services and integrations (p. 24).	July 9, 2015
Added service support	This release supports Amazon DynamoDB. See CloudTrail supported services and integrations (p. 24).	May 28, 2015

Change	Description	Release Date
Added service support	This release supports CloudWatch Logs in the US West (N. California) Region. See the CloudTrail release notes. For more information about CloudTrail support for CloudWatch Logs monitoring, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277).	May 19, 2015
Added service support	This release supports Amazon Directory Service. See CloudTrail supported services and integrations (p. 24).	May 14, 2015
Added service support	This release supports Amazon Simple Email Service (Amazon SES). See CloudTrail supported services and integrations (p. 24).	May 7, 2015
Added service support	This release supports Amazon Elastic Container Service See CloudTrail supported services and integrations (p. 24).	April 9, 2015
Added service support	This release supports Amazon Lambda. See CloudTrail supported services and integrations (p. 24).	April 9, 2015
Added service support	This release supports Amazon WorkSpaces. See CloudTrail supported services and integrations (p. 24).	April 9, 2015
	This release supports the lookup of Amazon activity captured by CloudTrail (CloudTrail events). You can look up and filter events in your account related to creation, modification, or deletion. To look up these events, you can use the CloudTrail console, the Amazon Command Line Interface (Amazon CLI), or the Amazon SDK. For more information, see Viewing events with CloudTrail Event history (p. 50).	March 12, 2015
Added service support and new documentation	This release supports Amazon CloudWatch Logs in the Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), and Europe (Frankfurt) Regions. For more information, see Sending events to CloudWatch Logs.	March 5, 2015
New documentation	A new section that describes CloudTrail support for Amazon Security Token Service (Amazon STS) regional endpoints has been added to the CloudTrail Concepts page.	February 17, 2015
Added service support	This release supports Amazon Route 53. See CloudTrail supported services and integrations (p. 24).	February 11, 2015
Added service support	This release supports Amazon Config. See CloudTrail supported services and integrations (p. 24).	February 10, 2015
Added service support	This release supports Amazon CloudHSM. See CloudTrail supported services and integrations (p. 24).	January 8, 2015
Added service support	This release supports Amazon CodeDeploy. See CloudTrail supported services and integrations (p. 24).	December 17, 2014
Added service support	This release supports Amazon Storage Gateway. See CloudTrail supported services and integrations (p. 24).	December 16, 2014

Change	Description	Release Date
Added Region support	This release supports one additional Region: us-gov-west-1 (Amazon GovCloud (US-West)). See CloudTrail supported Regions (p. 15).	December 16, 2014
Added service support	This release supports Amazon S3 Glacier. See CloudTrail supported services and integrations (p. 24).	December 11, 2014
Added service support	This release supports Amazon Data Pipeline. See CloudTrail supported services and integrations (p. 24).	December 2, 2014
Added service support	This release supports Amazon Key Management Service. See CloudTrail supported services and integrations (p. 24).	November 12, 2014
New documentation	A new section, Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 277), has been added to the guide. It describes how to use Amazon CloudWatch Logs to monitor CloudTrail log events.	November 10, 2014
New documentation	A new section, Using the CloudTrail Processing Library (p. 327), has been added to the guide. It provides information about how to write a CloudTrail log processor in Java using the Amazon CloudTrail Processing Library.	November 5, 2014
Added service support	This release supports Amazon Elastic Transcoder. See CloudTrail supported services and integrations (p. 24).	October 27, 2014
Added Region support	This release supports one additional region: eu-central-1 (Europe (Frankfurt)). See CloudTrail supported Regions (p. 15).	October 23, 2014
Added service support	This release supports Amazon CloudSearch. See CloudTrail supported services and integrations (p. 24).	October 16, 2014
Added service support	This release supports Amazon Simple Notification Service. See CloudTrail supported services and integrations (p. 24).	October 09, 2014
Added service support	This release supports Amazon ElastiCache. See CloudTrail supported services and integrations (p. 24).	September 15, 2014
Added service support	This release supports Amazon WorkDocs. See CloudTrail supported services and integrations (p. 24).	August 27, 2014
Added new content	This release includes a topic that discusses logging sign-in events. See Amazon Web Services Management Console sign-in events (p. 431).	July 24, 2014
Added new content	The **eventVersion** element for this release has been upgraded to version 1.02 and three new fields have been added. See CloudTrail record contents (p. 402).	July 18, 2014
Added service support	This release supports Auto Scaling (see CloudTrail supported services and integrations (p. 24)).	July 17, 2014

Change	Description	Release Date
Added Region support	This release supports three additional Regions: ap-southeast-1 (Asia Pacific (Singapore)), ap-northeast-1 (Asia Pacific (Tokyo)), sa-east-1 (South America (São Paulo)). See CloudTrail supported Regions (p. 15).	June 30, 2014
Additional service support	This release supports Amazon Redshift. See CloudTrail supported services and integrations (p. 24).	June 10, 2014
Added service support	This release supports Amazon OpsWorks. See CloudTrail supported services and integrations (p. 24).	June 5, 2014
Added service support	This release supports Amazon CloudFront. See CloudTrail supported services and integrations (p. 24).	May 28, 2014
Added Region support	This release supports three additional Regions: us-west-1 (US West (N. California)), eu-west-1 (Europe (Ireland)), ap-southeast-2 (Asia Pacific (Sydney)). See CloudTrail supported Regions (p. 15).	May 13, 2014
Added service support	This release supports Amazon Simple Workflow Service. See CloudTrail supported services and integrations (p. 24).	May 9, 2014
Added new content	This release includes topics that discuss sharing log files between accounts. See Sharing CloudTrail log files between Amazon accounts (p. 297).	May 2, 2014
Added service support	This release supports Amazon CloudWatch. See CloudTrail supported services and integrations (p. 24).	April 28, 2014
Added service support	This release supports Amazon Kinesis. See CloudTrail supported services and integrations (p. 24).	April 22, 2014
Added service support	This release supports Amazon Direct Connect. See CloudTrail supported services and integrations (p. 24).	April 11, 2014
Added service support	This release supports Amazon EMR. See CloudTrail supported services and integrations (p. 24).	April 4, 2014
Added service support	This release supports Elastic Beanstalk. See CloudTrail supported services and integrations (p. 24).	April 2, 2014
Additional service support	This release supports Amazon CloudFormation. See CloudTrail supported services and integrations (p. 24).	March 7, 2014
New guide	This release introduces Amazon CloudTrail.	November 13, 2013

Amazon glossary

For the latest Amazon terminology, see the Amazon glossary in the *Amazon Web Services Glossary Reference*.