

---

# **AWS CloudTrail**

**User Guide**

**Version 1.0**



## AWS CloudTrail: User Guide

Copyright © 2015 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, AWS CloudTrail, AWS CodeDeploy, Amazon Cognito, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Amazon Kinesis, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC, and Amazon WorkDocs. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

## Table of Contents

What is AWS CloudTrail? .....	1
How AWS CloudTrail Works .....	1
CloudTrail Concepts .....	2
Creating a Trail .....	2
CloudTrail Console .....	2
CloudTrail CLI .....	2
CloudTrail APIs .....	2
AWS SDKs .....	3
IAM and CloudTrail .....	3
CloudWatch Logs and CloudTrail .....	3
Regional and Global Services .....	3
How Does CloudTrail Relate to Other AWS Monitoring Services? .....	4
Partner Solutions .....	4
CloudTrail Workflow .....	4
CloudTrail Supported Services .....	5
Analytics .....	5
Application Services .....	6
Compute .....	7
Database .....	7
Developer Tools .....	8
Enterprise Applications .....	8
Management Tools .....	9
Mobile Services .....	9
Networking .....	10
Security and Identity .....	10
Storage and Content Delivery .....	11
CloudTrail Topics by AWS Service .....	11
CloudTrail Supported Regions .....	13
CloudTrail Log File Examples .....	14
CloudTrail Log File Name Format .....	14
Log Examples .....	15
Getting Started with CloudTrail .....	21
Creating and Updating Your Trail .....	21
Amazon S3 Bucket Naming Requirements .....	22
Amazon S3 Bucket Policy for CloudTrail .....	22
Creating and Updating a Trail with the CloudTrail Console .....	26
Creating and Updating a Trail with the AWS CLI .....	27
Viewing CloudTrail Events .....	30
Viewing CloudTrail Events in the CloudTrail Console .....	31
Viewing CloudTrail Events with the AWS CLI .....	32
Regions Supported for Event Lookup in CloudTrail .....	37
Services Supported for Event Lookup in CloudTrail .....	38
APIs Supported for Event Lookup in CloudTrail .....	39
Resource Types Supported for Event Lookup in CloudTrail .....	60
Controlling User Permissions for CloudTrail .....	63
Granting Permissions for CloudTrail Administration .....	64
Granting Custom Permissions for CloudTrail Users .....	65
Getting and Viewing Your CloudTrail Log Files .....	67
Finding Your CloudTrail Log Files .....	67
Reading Your CloudTrail Log Files .....	68
Configuring Amazon SNS Notifications for CloudTrail .....	69
Configuring CloudTrail to Send Notifications .....	69
CloudTrail Permissions for SNS Notifications .....	70
Working with CloudTrail Log Files .....	72
Receiving Log Files in a Single Amazon S3 Bucket .....	72

Receiving CloudTrail Log Files from Multiple Regions .....	72
Receiving CloudTrail Log Files from Multiple Accounts .....	78
Sharing CloudTrail Log Files Between AWS Accounts .....	81
Scenario 1: Granting Access to the Account that Generated the Log Files .....	82
Scenario 2: Granting Access to All Logs .....	84
Creating a Role .....	85
Creating an Access Policy to Grant Access to Accounts You Own .....	86
Creating an Access Policy to Grant Access to a Third Party .....	88
Assuming a Role .....	89
Stop Sharing CloudTrail Log Files Between AWS Accounts .....	91
Monitoring Log Files with Amazon CloudWatch Logs .....	92
Sending CloudTrail Events to CloudWatch Logs .....	92
Using an AWS CloudFormation Template to Create CloudWatch Alarms .....	95
Creating CloudWatch Alarms for CloudTrail Events: Examples .....	104
Creating CloudWatch Alarms for CloudTrail Events: Additional Examples .....	127
Configuring Notifications for CloudWatch Logs Alarms .....	133
Stopping CloudTrail from Sending Events to CloudWatch Logs .....	133
CloudWatch Log Group and Log Stream Naming for CloudTrail .....	134
Role Policy Document for CloudTrail to Use CloudWatch Logs for Monitoring .....	134
Using the CloudTrail Processing Library .....	135
Minimum requirements .....	135
Processing CloudTrail Logs with the CloudTrail Processing Library .....	135
Advanced Topics .....	139
Additional Resources .....	142
CloudTrail Event Reference .....	143
CloudTrail Record Contents .....	144
CloudTrail userIdentity Element .....	146
Examples .....	146
Fields .....	147
Tracing Federated Identities in CloudTrail Logs .....	149
Characteristics of an Identity Broker Application .....	149
Non-API Events Captured by CloudTrail .....	150
AWS Console Sign-in Events .....	151
CloudTrail Document History .....	154

# What is AWS CloudTrail?

---

Welcome to the *AWS CloudTrail User Guide*. With AWS CloudTrail you can get a history of AWS API calls and related events for your account. This includes calls made by using the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services.

You can identify which users and accounts called AWS for services that support CloudTrail, the source IP address the calls were made from, and when the calls occurred. You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of your trails, and control how administrators turn CloudTrail logging on and off.

## Topics

- [How AWS CloudTrail Works](#) (p. 1)
- [CloudTrail Concepts](#) (p. 2)
- [CloudTrail Workflow](#) (p. 4)
- [CloudTrail Supported Services](#) (p. 5)
- [CloudTrail Supported Regions](#) (p. 13)
- [CloudTrail Log File Examples](#) (p. 14)

## How AWS CloudTrail Works

AWS CloudTrail captures AWS API calls and related events made by or on behalf of an AWS account and delivers log files to an Amazon S3 bucket that you specify. Using the CloudTrail console, the AWS CLI, or the CloudTrail API, you create a *trail*, which specifies the bucket for log file delivery and storage. By default, your log files are encrypted using Amazon S3 server-side encryption (SSE). You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. You can also optionally configure AWS CloudTrail to deliver events to a log group to be monitored by CloudWatch Logs.

CloudTrail typically delivers log files within 15 minutes of an API call. In addition, the service publishes new log files multiple times an hour, usually about every five minutes. These log files contain API calls from all of the account's services that support CloudTrail. For a list of AWS services that support CloudTrail, see [CloudTrail Supported Services](#) (p. 5).

### Note

AWS CloudTrail records API calls made on an AWS account directly by the user or on behalf of the user by an AWS service. Examples of services that make API calls on behalf of users include, but are not limited to, AWS CloudFormation, Elastic Beanstalk, AWS OpsWorks, and Auto

Scaling. For example, a AWS CloudFormation `CreateStack` call can result in additional API calls to Amazon EC2, Amazon RDS, Amazon EBS or other services as prescribed in the AWS CloudFormation template. This behavior is normal and expected. CloudTrail logs all of these API calls and provides a history of the calls made by the users directly or made by an AWS service as a result of the calls made by the user. You can identify the latter type of API call by examining the **invokedBy** field in the CloudTrail event.

If you want to take quick action upon log file delivery, you can choose to have CloudTrail publish SNS notifications when new log files are delivered. For information, see [Configuring Amazon SNS Notifications for CloudTrail](#) (p. 69). You can also choose to have CloudTrail receive SNS notifications that you configured for CloudWatch alarms.

You can aggregate log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For information, see [Receiving CloudTrail Log Files from Multiple Sources in a Single Amazon S3 Bucket](#) (p. 72). Similarly, you can aggregate CloudTrail events across all regions to a CloudWatch Logs log group in one region. However, you cannot aggregate CloudTrail events from different AWS accounts into one CloudWatch Logs log group belonging to one AWS account.

There is no additional charge for CloudTrail, but standard rates for Amazon S3 and Amazon SNS apply. For more information about Amazon S3 rates, see [Amazon S3 Pricing](#). For more information about Amazon SNS rates, see [Amazon SNS Pricing](#).

## CloudTrail Concepts

This section summarizes basic concepts related to CloudTrail.

### Creating a Trail

Creating a trail means setting the configuration options to start logging AWS API calls and related events. That is, you must turn on the CloudTrail service, set up the target Amazon S3 bucket, (optionally) set up a log group for CloudWatch Logs to monitor log events, and (optionally) create an Amazon SNS topic to deliver CloudTrail notifications to you.

### CloudTrail Console

The AWS CloudTrail console is a web application that you can use to manage the CloudTrail service. The console provides a user interface for performing many CloudTrail tasks such as turning on or editing CloudTrail, selecting an Amazon S3 bucket, setting a prefix, including or preventing API calls from global services such as IAM and AWS STS, and receiving Amazon SNS notifications for log file deliveries. For more information about the AWS management console in general, see [AWS Management Console](#) .

### CloudTrail CLI

The AWS Command Line Interface is a unified tool that enables you to act easily with CloudTrail from the command line. For more information, see the [CLI User Guide](#) . For a complete list of the available 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 [AWS CloudTrail API Reference](#) .

## AWS SDKs

As an alternative to using the CloudTrail API, you can use one of the AWS 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, the SDKs take care of cryptographically signing requests, managing errors, and retrying requests automatically. For more information, see the [Tools For AWS](#) page.

## IAM and CloudTrail

AWS Identity and Access Management is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions. Without IAM, organizations with multiple users and systems must either create multiple AWS accounts, each with its own billing and subscriptions to AWS products, or employees must all share the security credentials of a single AWS account. Also, without IAM, you have no control over the tasks a particular user or system can do and what AWS resources they might use.

Use IAM to create individual users for anyone who needs access to AWS CloudTrail. Create an IAM user for yourself as well, give that IAM user administrative privileges, and use that IAM user for all your work. By creating individual IAM users for people accessing your account, you can give each IAM user a unique set of security credentials. You can also grant different permissions to each IAM user. If necessary, you can change or revoke an IAM user's permissions any time. For more information, see [Controlling User Permissions for CloudTrail](#) (p. 63).

## CloudWatch Logs and CloudTrail

Amazon CloudWatch is a web service that collects and tracks metrics to monitor in real time your Amazon Web Services (AWS) resources and the applications that you run on Amazon Web Services (AWS). 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 AWS 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 AWS services that are sent to CloudWatch Logs.

## Regional and Global Services

For most services, events are sent to the region where the action happened. For the IAM, AWS STS, CloudFront, and AWS Management Console services, events are delivered to any trail that has the **Include global services** option enabled.

### CloudTrail is a Regional Service

CloudTrail is a regional service and you enable it per region. The trail that you create in each region delivers only events that occur in that region. However, you can include events from the global services listed above by checking the **Include global services** flag. This option is enabled by default when you create the trail for a region.

For convenience, you can configure CloudTrail to use the same Amazon S3 bucket for multiple regions. If you do this and leave the **Include global services** flag enabled in the trail for each region, global service events are duplicated in the logs. To prevent duplication, you can include global services in one trail and exclude them from the others. For more information, see [Turning On CloudTrail in Additional Regions](#) (p. 74).

**Note**

Global service events originate in the us-east-1 region, so if you specify a single Amazon S3 bucket for trail information from multiple regions in your account and are not using region-specific endpoints, consider creating the bucket in us-east-1. To avoid receiving duplicate events, you can then disable global services in the regions other than us-east-1.

## AWS Security Token Service (AWS STS) and CloudTrail

AWS STS is a global service that 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 AWS CloudTrail service. Regional endpoints help reduce latency in your applications.

When you use an AWS STS region-specific endpoint, the trail in that region delivers only the AWS 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 AWS STS events that originate from us-west-2. For more information about AWS STS regional endpoints, see [Activating AWS STS in a New Region](#).

For a complete list of AWS web service regional endpoints, see [Regions and Endpoints](#).

## How Does CloudTrail Relate to Other AWS Monitoring Services?

CloudTrail adds another dimension to the monitoring capabilities already offered by AWS; it does not change or replace logging features you might already be using. For example, where Amazon CloudWatch focuses on performance monitoring and system health, CloudTrail focuses on API activity. CloudTrail does not report on system performance or health, nor does it alter how you receive logs from your Amazon S3 or Amazon CloudFront subscriptions.

## Partner Solutions

AWS partners with third-party specialists in logging and analysis to provide solutions that leverage CloudTrail output. For more information, visit the CloudTrail detail page at [AWS CloudTrail](#).

## CloudTrail Workflow

Here are the steps you take to use CloudTrail, which are described in detail in this guide:

1. Using the console, the AWS CLI, or the CloudTrail API, you create a trail, which consists of the information that CloudTrail uses to deliver log files to your Amazon S3 bucket or CloudWatch Logs log group. For information, see [Creating and Updating Your Trail](#) (p. 21).
2. (Optional) You create an Amazon SNS topic to which you subscribe for notifications that a new log file has arrived in your bucket. Amazon SNS can notify you in multiple ways, including programmatically using Amazon Simple Queue Service. For information, see [Configuring Amazon SNS Notifications for CloudTrail](#) (p. 69).
3. You use the Amazon S3 API or console to retrieve the log files. For information, see [Getting and Viewing Your CloudTrail Log Files](#) (p. 67).
4. You use the CloudTrail API, AWS CLI, or console to update your trail.
5. (Optional) You can use AWS Identity and Access Management to control which AWS users can create, configure, or delete trails, start and stop logging, and access the buckets that contain log information. For information, see [Controlling User Permissions for CloudTrail](#) (p. 63).



6. (Optional) You can use CloudWatch Logs to monitor events from API activity captured by CloudTrail. If so, you also create an IAM role with permissions for CloudTrail to send events to a CloudWatch Logs log group for monitoring. For more information, see [\[Monitoring CloudTrail Log Files by Using Amazon CloudWatch Logs\]](#).
7. (Optional) You can analyze your CloudTrail output by using one of the partner solutions that integrate with CloudTrail. These solutions offer a broad set of capabilities, such as change tracking, troubleshooting, and security analysis. For more information, visit the [Amazon CloudTrail](#) page.

## CloudTrail Supported Services

This topic lists the services that currently support CloudTrail.

### Service Categories

- [Analytics](#) (p. 5)
- [Application Services](#) (p. 6)
- [Compute](#) (p. 7)
- [Database](#) (p. 7)
- [Developer Tools](#) (p. 8)
- [Enterprise Applications](#) (p. 8)
- [Management Tools](#) (p. 9)
- [Mobile Services](#) (p. 9)
- [Networking](#) (p. 10)
- [Security and Identity](#) (p. 10)
- [Storage and Content Delivery](#) (p. 11)
- [CloudTrail Topics by AWS Service](#) (p. 11)

## Analytics

- **Amazon Elastic MapReduce** (Supported beginning 04/04/2014)

Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several services from AWS to perform such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing. For more information, see the [Amazon EMR Developer Guide](#). For more information about the Amazon EMR calls logged by CloudTrail, see [Logging Amazon Elastic MapReduce API Calls in AWS CloudTrail](#).

- **AWS Data Pipeline** (Supported beginning 12/02/2014)

AWS Data Pipeline is a web service that you can use to automate the movement and transformation of data through data-driven workflows. For more information, see the [AWS Data Pipeline Developer Guide](#). For more information about the AWS Data Pipeline calls logged by CloudTrail, see [Logging AWS Data Pipeline API Calls by using AWS CloudTrail](#).

- **Amazon Kinesis** (Supported beginning 04/25/2014)

Amazon Kinesis is a managed service that scales elastically for real-time processing of streaming big data. The service takes in large streams of data records that can then be consumed in real time by multiple data-processing applications that can be run on Amazon EC2 instances. For more information, see the [Amazon Kinesis Developer Guide](#). For more information about the Amazon Kinesis calls logged by CloudTrail see [Logging Amazon Kinesis API Calls by using AWS CloudTrail](#).

- **Amazon Redshift** (Supported beginning 06/10/2014)

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data by using your existing business intelligence tools. It is optimized for datasets that range from a few hundred gigabytes to a petabyte or more. An Amazon Redshift data warehouse is a collection of computing resources called nodes which are organized into groups called clusters. Each cluster runs an Amazon Redshift engine and contains one or more databases. For more information, see the [Amazon Redshift Developer Guide](#). For a complete list of the Amazon Redshift actions logged by CloudTrail, see the [Amazon Redshift API Documentation](#).

## Application Services

- **Amazon API Gateway** (Supported beginning 07/09/2015)

Amazon API Gateway helps developers deliver robust, reliable, secure and scalable access to backend APIs for mobile apps, web apps, and server apps. For more information, see the [API Gateway Developer Guide](#). For more information about the Amazon API Gateway calls logged by CloudTrail, see [Logging Amazon API Gateway API Calls By Using AWS CloudTrail](#).

- **Amazon CloudSearch** (Supported beginning 10/16/2014)

Amazon CloudSearch is a fully-managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website. Amazon CloudSearch enables you to search large collections of data such as web pages, document files, forum posts, or product information. For more information about Amazon CloudSearch, see the [Amazon CloudSearch Developer Guide](#). For more information about the Amazon CloudSearch calls logged by CloudTrail, see [Logging Amazon CloudSearch Configuration Service Calls Using AWS CloudTrail](#).

- **Amazon Elastic Transcoder** (Supported beginning 10/27/2014)

Amazon Elastic Transcoder lets you convert media files that you have stored in Amazon S3 into media files in the formats required by consumer playback devices. For more information about Elastic Transcoder, see [Amazon Elastic Transcoder Developer Guide](#). For more information about the Elastic Transcoder calls logged by CloudTrail, see [Logging Elastic Transcoder API Calls Using CloudTrail](#).

- **Amazon Simple Email Service** (Supported beginning 05/07/2015)

Amazon Simple Email Service is an outbound-only email-sending service that provides an easy, cost-effective way for you to send email. For more information about Amazon Simple Email Service, see the [Amazon Simple Email Service Developer Guide](#). For more information about the Amazon SES calls logged by CloudTrail, see [Logging Amazon SES API Calls By Using AWS CloudTrail](#).

- **Amazon Simple Queue Service** (Supported beginning 07/16/2014)

Amazon Simple Queue Service (Amazon SQS) offers reliable and scalable hosted queues for storing messages as they travel between computers. By using Amazon SQS, you can move data between distributed components of your applications that perform different tasks without losing messages or requiring each component to be always available. For more information, see the [Amazon Simple Queue Service Developer Guide](#). For more information about the Amazon SQS calls logged by CloudTrail, see [Logging Amazon SQS API Calls By Using AWS CloudTrail](#).

- **Amazon Simple Workflow Service** (Supported beginning 05/13/2014)

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that coordinate work across distributed components. In Amazon SWF, a task represents a logical unit of work that is performed by a component of your application. Coordinating tasks across the application involves managing intertask dependencies, scheduling, and concurrency in accordance with the logical flow of the application. Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state. For more information, see the [Amazon SWF Developer Guide](#). For more information about the Amazon SWF calls logged by CloudTrail, see [Logging Amazon Simple Workflow Service API Calls with AWS CloudTrail](#).

## Compute

- **Amazon Elastic Compute Cloud (EC2)** (Supported beginning 11/13/2013)

Amazon Elastic Compute Cloud (Amazon EC2) provides resizable computing capacity in the AWS cloud. You can launch as many or as few virtual servers as you need, 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 about Amazon EC2, see the [Amazon EC2 User Guide](#). For more information about the Amazon EC2 calls logged by CloudTrail, see [Logging Amazon EC2 API Calls Using AWS CloudTrail](#). [Amazon EC2 Simple Systems Manager \(SSM\)](#) is a feature of [EC2Config](#) that enables you to manage the configuration of running Windows instances. For information about the Amazon EC2 Simple Systems Manager API calls logged by CloudTrail, see [Logging SSM API Calls Using AWS CloudTrail](#).

- **Amazon EC2 Container Service** (Supported beginning 04/09/2015)

Amazon EC2 Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances. For more information about Amazon EC2 Container Service, see the [Amazon EC2 Container Service Developer Guide](#). For more information about the Amazon ECS calls logged by CloudTrail, see [Logging Amazon ECS API Calls By Using AWS CloudTrail](#).

- **AWS Elastic Beanstalk** (Supported beginning 03/31/2014)

You can use Elastic Beanstalk to quickly deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications. For more information, see the [Elastic Beanstalk Developer's Guide](#). For more information about the Elastic Beanstalk calls logged by using CloudTrail, see [Using AWS Elastic Beanstalk with AWS CloudTrail](#).

- **AWS Lambda** (Supported beginning 04/09/2015)

AWS Lambda is a zero-administration compute platform that runs your code in the AWS cloud, providing the high availability, security, performance, and scalability of AWS infrastructure. For more information about Lambda, see the [AWS Lambda Developer Guide](#). For more information about the Lambda calls logged by CloudTrail, see [Logging AWS Lambda API Calls By Using AWS CloudTrail](#).

- **Auto Scaling** (Supported beginning 07/16/2014)

Auto Scaling is a web service that enables you to automatically launch or terminate Amazon Elastic Compute Cloud (Amazon EC2) instances based on user-defined policies, health status checks, and schedules. For more information, see the [Auto Scaling Developer Guide](#). For a list of the Auto Scaling calls logged by CloudTrail, see [Logging Auto Scaling API Calls By Using CloudTrail](#).

- **Elastic Load Balancing** (Supported beginning 04/04/2014)

You can use Elastic Load Balancing to automatically distribute your incoming application traffic across multiple Amazon EC2 instances. Elastic Load Balancing automatically scales request handling capacity in response to incoming traffic. For more information about Elastic Load Balancing, see the [Elastic Load Balancing Developer Guide](#). For more information about the Elastic Load Balancing calls logged by CloudTrail, see [Logging Elastic Load Balancing API Calls Using AWS CloudTrail](#).

## Database

- **Amazon Relational Database Service** (Supported beginning 11/13/2013)

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. For more information, see the [Amazon RDS User Guide](#). For more information about the Amazon RDS calls logged by CloudTrail, see [Logging Amazon RDS API Calls Using CloudTrail](#).

- **Amazon DynamoDB** (Supported beginning 05/28/2015)

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. For more information, see the [Amazon DynamoDB Developer Guide](#). For more information about the DynamoDB calls logged by CloudTrail, see [Logging DynamoDB API Calls By Using AWS CloudTrail](#).

- **Amazon ElastiCache** (Supported beginning 09/15/2014)

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud. It provides a high performance, resizeable, and cost-effective in-memory cache, while removing the complexity associated with deploying and managing a distributed cache environment. For more information, see the [Amazon ElastiCache User Guide](#). For more information about the ElastiCache calls logged by CloudTrail, see [Logging Amazon ElastiCache API Calls Using AWS CloudTrail](#).

## Developer Tools

- **AWS CodeDeploy** (Supported beginning 12/16/2014)

AWS CodeDeploy is a deployment service that enables developers to automate the deployment of applications to Amazon Elastic Compute Cloud (Amazon EC2) instances, and to update the applications as required. For more information, see the [AWS CodeDeploy User Guide](#). For more information about the AWS CodeDeploy calls logged by CloudTrail, see [Logging CodeDeploy API Calls By Using AWS CloudTrail](#).

- **AWS CodePipeline** (Supported beginning 07/09/2015)

AWS CodePipeline is a continuous delivery and automation service hosted by Amazon Web Services that enables you to model, configure, and automate the steps required to release your software. For more information, see the [AWS CodePipeline User Guide](#). For more information about the AWS CodePipeline calls logged by CloudTrail, see [Logging AWS CodePipeline API Calls By Using AWS CloudTrail](#).

## Enterprise Applications

- **Amazon WorkSpaces** (Supported beginning 04/09/2015)

Amazon WorkSpaces offers an easy way to provide a cloud-based desktop experience to your end-users. A choice of bundles offer a range of different amounts of CPU, memory, storage, and a choice of applications. Users can connect from a PC, Mac desktop computer, iPad, Kindle, or Android tablet. For more information about Amazon WorkSpaces, see [Amazon WorkSpaces Administration Guide](#). For more information about the Amazon WorkSpaces actions logged by CloudTrail, see [Logging Amazon WorkSpaces API Calls by Using CloudTrail](#).

- **Amazon WorkDocs** (Supported beginning 08/27/2014)

Amazon WorkDocs is a fully managed enterprise storage and sharing service. Your files are stored in the cloud safely and securely. Amazon WorkDocs also includes a synchronization application that keeps selected folders on your local computer in sync with your files in the cloud. Your files are visible to only you and to your designated contributors and viewers. For more information about Amazon WorkDocs, see [Amazon WorkDocs Administration Guide](#). For more information about the Amazon WorkDocs actions logged by CloudTrail, see [Logging Amazon WorkDocs API Calls By Using CloudTrail](#).

## Management Tools

- **Amazon CloudWatch** (Supported beginning 04/30/2014)

Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics which are the variables you want to measure for your resources and applications. CloudWatch alarms send notifications or automatically make changes to the resources you are monitoring based on rules that you define. For more information about CloudWatch, see the [Amazon CloudWatch Developer Guide](#). For more information about the list of CloudWatch calls logged by CloudTrail, see [Logging Amazon CloudWatch API Calls in AWS CloudTrail](#).

- **AWS CloudFormation** (Supported beginning 04/02/2014)

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products such as Amazon EC2, Amazon EBS, Amazon SNS, Elastic Load Balancing, and Auto Scaling to build highly reliable, highly scalable, cost-effective applications without worrying about creating and configuring the underlying AWS infrastructure. For more information, see the [AWS CloudFormation User Guide](#). For more information about the AWS CloudFormation calls logged by CloudTrail, see [Logging AWS CloudFormation API Calls in AWS CloudTrail](#).

- **AWS CloudTrail** (Supported beginning 11/13/2013)

Like any supported service, when logging is turned on, CloudTrail logs actions to an Amazon S3 bucket that you specify. For a complete list of the actions that can be logged, see the [CloudTrail API Reference](#).

- **AWS Config** (Supported beginning 02/10/2015)

AWS Config provides a detailed view of the resources associated with your AWS account, including how they are configured, how they are related to one another, and how the configurations and their relationships have changed over time. For more information, see the [AWS Config Developer Guide](#). For information about the AWS Config calls logged by CloudTrail, see [Logging AWS Config API Calls By Using AWS CloudTrail](#).

- **AWS OpsWorks** (Supported beginning 06/04/2014)

AWS OpsWorks provides a simple and flexible way to create and manage stacks and applications. It supports a standard set of components—including application servers, database servers, load balancers, and more—that you can use to assemble your stack. These components all come with a standard configuration and are ready to run. For more information, see the [AWS OpsWorks User Guide](#). For more information about the list of AWS OpsWorks calls logged by CloudTrail, see [Logging AWS OpsWorks API Calls By Using AWS CloudTrail](#).

## Mobile Services

- **AWS Device Farm** (Supported beginning 07/13/2015)

AWS Device Farm is an app testing service that enables you to test your Android and Fire OS apps on real, physical phones and tablets that are hosted by Amazon Web Services (AWS). For more information about AWS Device Farm, see the [Device Farm Developer Guide](#). For information about the AWS Device Farm calls logged by CloudTrail, see [Logging AWS Device Farm API Calls By Using AWS CloudTrail](#).

- **Amazon Simple Notification Service** (Supported beginning 10/09/2014)

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. For more information about Amazon SNS, see the [Amazon Simple Notification Service Developer Guide](#). For more information about the Amazon SNS calls logged by CloudTrail, see [Logging Amazon Simple Notification Service API Calls By Using AWS CloudTrail](#).

## Networking

- **Amazon Virtual Private Cloud** (Supported beginning 11/13/2013)

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you would operate in your own data center with the added benefit of using the scalable AWS infrastructure. For more information, see the [Amazon Virtual Private Cloud User Guide](#). The Amazon VPC API is a subset of the Amazon EC2 API. For more information about the Amazon EC2 calls logged by CloudTrail (including those for Amazon VPC), see [Logging Amazon EC2 API Calls Using AWS CloudTrail](#).

- **AWS Direct Connect** (Supported beginning 03/08/2014)

You can use AWS Direct Connect to establish a direct connection from your premises to AWS. This may reduce your network costs and increase bandwidth throughput. For more information about AWS Direct Connect, see the [AWS Direct Connect User Guide](#). For more information about the AWS Direct Connect calls logged by CloudTrail, see [Logging AWS Direct Connect API Calls in AWS CloudTrail](#).

- **Amazon Route 53** (Supported beginning 02/11/2015)

Amazon Route 53 is a Domain Name System (DNS) and domain name registration web service. To use Amazon Route 53 with CloudTrail, you must choose US East (N. Virginia) as the region when you create the trail. For more information about Amazon Route 53, see the [Amazon Route 53 Developer Guide](#). For information about the Amazon Route 53 calls logged by CloudTrail, see [Using AWS CloudTrail to Capture Requests Sent to the Amazon Route 53 API](#).

## Security and Identity

- **AWS Identity and Access Management** (Supported beginning 11/13/2013)

AWS Identity and Access Management (IAM) is a web service that enables AWS customers to manage users and user permissions. By using IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access. For more information, see the [IAM User Guide](#). For more information about the IAM calls logged by CloudTrail, see [Logging IAM Events with AWS CloudTrail](#).

- **AWS CloudHSM** (Supported beginning 01/08/2015)

AWS CloudHSM provides secure cryptographic key storage to customers by making hardware security modules (HSMs) available in the AWS cloud. For more information, see the [AWS CloudHSM Getting Started Guide](#). For a complete list of AWS CloudHSM calls logged by CloudTrail, see [Logging AWS CloudHSM API Calls By Using AWS CloudTrail](#).

- **AWS Directory Service** (Supported beginning 05/14/2015)

The AWS Directory Service is a managed service that makes it easy to connect to your existing on-premises Microsoft Active Directory and deploy and manage Windows workloads in the AWS cloud. For more information about AWS Directory Service, see the [AWS Directory Service Administration Guide](#). For information about the AWS Directory Service calls logged by CloudTrail, see [Logging AWS Directory Service API Calls by Using CloudTrail](#).

- **AWS Key Management Service** (Supported beginning 11/12/2014)

The AWS Key Management Service is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS KMS is integrated with other AWS services including Amazon EBS, Amazon S3, and Amazon Redshift. For more information, see the [AWS Key Management Service Developer Guide](#). For more information about the list of AWS KMS calls logged by CloudTrail, see [Logging AWS KMS API Calls](#).

- **AWS Security Token Service** (Supported beginning 11/13/2013)

You can use the AWS Security Token Service (STS) to grant a trusted user temporary, limited access to your AWS resources. For more information, see the [AWS Security Token Service User Guide](#). For information about both the IAM and AWS STS calls logged by CloudTrail, see [Logging IAM Events with AWS CloudTrail](#). For a complete list of AWS STS calls, see the [AWS Security Token Service API Reference](#).

## Storage and Content Delivery

- **Amazon S3 bucket level events** (Supported beginning 09/01/2015)

You can use Amazon Simple Storage Service (Amazon S3) to store and retrieve any amount of data at any time, from anywhere on the web. For more information, see the [Amazon Simple Storage Service Developer Guide](#). With this release, CloudTrail logs Amazon S3 bucket-related events such as the creation and deletion of buckets, changes to bucket policy, and changes to replication status. You can also use CloudTrail logs together with Amazon S3 server access logs. For detailed information, see [Logging Amazon S3 API Calls By Using AWS CloudTrail](#).

- **Amazon CloudFront** (Supported beginning 05/28/2014)

Amazon CloudFront speeds up distribution of your static and dynamic web content to end users. CloudFront delivers your content through a worldwide network of edge locations. When an end user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency, so that content is delivered with the best possible performance. For more information, see the [Amazon CloudFront Developer Guide](#). For more information about the CloudFront calls logged by CloudTrail, see [Using AWS CloudTrail to Capture Requests Sent to the CloudFront API](#).

- **Amazon Glacier** (Supported beginning 12/11/2014)

Amazon Glacier is a storage service optimized for data archiving and backup of infrequently used data. The service is durable, extremely low-cost, and includes security features. For more information, see the [Amazon Glacier Developer Guide](#). For more information about the Amazon Glacier calls logged by CloudTrail, see [Logging Amazon Glacier API Calls By Using AWS CloudTrail](#).

- **AWS Storage Gateway** (Supported beginning 12/16/2014)

AWS Storage Gateway is a service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between your on-premises IT environment and the AWS storage infrastructure in the cloud. For more information about AWS Storage Gateway, see the [AWS Storage Gateway User Guide](#). For information about the AWS Storage Gateway Volume Gateway calls logged by CloudTrail, see [Logging Volume Gateway API Calls by Using AWS CloudTrail](#). For information about the Gateway-VTL calls logged by CloudTrail, see [Logging Gateway-VTL API Calls by Using AWS CloudTrail](#).

## CloudTrail Topics by AWS Service

This page provides links to CloudTrail topics for specific AWS services.

<b>AWS Service</b>	<b>CloudTrail Topics</b>
Amazon API Gateway	<a href="#">Logging Amazon API Gateway API Calls By Using AWS CloudTrail</a>
Auto Scaling	<a href="#">Logging Auto Scaling API Calls By Using CloudTrail</a>
AWS CloudFormation	<a href="#">Logging AWS CloudFormation API Calls in AWS CloudTrail</a>
Amazon CloudFront	<a href="#">Using AWS CloudTrail to Capture Requests Sent to the CloudFront API</a>
AWS CloudHSM	<a href="#">Logging AWS CloudHSM API Calls By Using AWS CloudTrail</a>

**AWS CloudTrail User Guide**  
**CloudTrail Topics by AWS Service**

<b>AWS Service</b>	<b>CloudTrail Topics</b>
Amazon CloudSearch	<a href="#">Logging Amazon CloudSearch Configuration Service Calls Using AWS CloudTrail</a>
Amazon CloudWatch	<a href="#">Logging Amazon CloudWatch API Calls in AWS CloudTrail</a>
AWS CodeDeploy	<a href="#">Logging CodeDeploy API Calls By Using AWS CloudTrail</a>
AWS CodePipeline	<a href="#">Logging AWS CodePipeline API Calls By Using AWS CloudTrail.</a>
AWS Config	<a href="#">Logging AWS Config API Calls By Using AWS CloudTrail</a>
AWS Data Pipeline	<a href="#">Logging AWS Data Pipeline API Calls by using AWS CloudTrail</a>
AWS Device Farm	<a href="#">Logging AWS Device Farm API Calls By Using AWS CloudTrail</a>
AWS Direct Connect	<a href="#">Logging AWS Direct Connect API Calls in AWS CloudTrail</a>
AWS Directory Service	<a href="#">Logging AWS Directory Service API Calls by Using CloudTrail</a>
Amazon DynamoDB	<a href="#">Logging DynamoDB API Calls By Using AWS CloudTrail</a>
Amazon Elastic Compute Cloud (Amazon EC2)	<a href="#">Logging Amazon EC2 API Calls Using AWS CloudTrail</a>
Amazon EC2 Container Service (Amazon ECS)	<a href="#">Logging Amazon ECS API Calls By Using AWS CloudTrail</a>
Elastic Load Balancing	<a href="#">Logging Elastic Load Balancing API Calls Using AWS CloudTrail</a>
Amazon Elastic MapReduce (Amazon EMR)	<a href="#">Logging Amazon Elastic MapReduce API Calls in AWS CloudTrail</a>
Amazon Elastic Transcoder	<a href="#">Logging Elastic Transcoder API Calls Using CloudTrail</a>
Amazon ElastiCache	<a href="#">Logging Amazon ElastiCache API Calls Using AWS CloudTrail</a>
Amazon Glacier	<a href="#">Logging Amazon Glacier API Calls By Using AWS CloudTrail</a>
AWS Identity and Access Management (IAM)	<a href="#">Logging IAM Events with AWS CloudTrail</a>
Amazon Kinesis	<a href="#">Logging Amazon Kinesis API Calls by using AWS CloudTrail</a>
AWS Key Management Service (AWS KMS)	<a href="#">Logging AWS KMS API Calls</a>
AWS Lambda	<a href="#">Logging AWS Lambda API Calls By Using AWS CloudTrail</a>  <a href="#">AWS Lambda Walkthrough 5: Handling AWS CloudTrail Events Using the AWS CLI (Node.js)</a>
AWS OpsWorks	<a href="#">Logging AWS OpsWorks API Calls By Using AWS CloudTrail</a>
Amazon Relational Database Service (Amazon RDS)	<a href="#">Logging Amazon RDS API Calls Using CloudTrail</a>
Amazon Route 53	<a href="#">Using AWS CloudTrail to Capture Requests Sent to the Amazon Route 53 API</a>



<b>AWS Service</b>	<b>CloudTrail Topics</b>
Amazon S3 Bucket Level Events	<a href="#">Logging Amazon S3 API Calls By Using AWS CloudTrail</a>
Amazon Simple Email Service (Amazon SES)	<a href="#">Logging Amazon SES API Calls By Using AWS CloudTrail</a>
Amazon Simple Notification Service (Amazon SNS)	<a href="#">Logging Amazon Simple Notification Service API Calls By Using AWS CloudTrail</a>
Amazon Simple Queue Service (Amazon SQS)	<a href="#">Logging Amazon SQS API Calls By Using AWS CloudTrail</a>
Amazon EC2 Simple Systems Manager (SSM)	<a href="#">Logging SSM API Calls Using AWS CloudTrail</a>
Amazon Simple Workflow Service (Amazon SWF)	<a href="#">Logging Amazon Simple Workflow Service API Calls with AWS CloudTrail</a>
AWS Storage Gateway	<a href="#">Logging Volume Gateway API Calls by Using AWS CloudTrail</a> <a href="#">Logging Gateway-VTL API Calls by Using AWS CloudTrail</a>
AWS Security Token Service (AWS STS)	<a href="#">Logging IAM Events with AWS CloudTrail</a> The IAM topic includes information for AWS STS.
Amazon Virtual Private Cloud (Amazon VPC)	<a href="#">Logging Amazon EC2 API Calls Using AWS CloudTrail</a> The Amazon VPC API is a subset of the Amazon EC2 API.
Amazon WorkDocs	<a href="#">Logging Amazon WorkDocs API Calls By Using CloudTrail</a>
Amazon WorkSpaces	<a href="#">Logging Amazon WorkSpaces API Calls by Using CloudTrail</a>

## CloudTrail Supported Regions

AWS CloudTrail currently supports the following regions:

<b>Region Name</b>	<b>Region</b>	<b>Endpoint</b>	<b>Protocol</b>	<b>AWS Account ID</b>	<b>Support Date</b>	<b>Integration with Cloud-Watch Logs</b>
Asia Pacific (Tokyo)	ap-north-east-1	cloudtrail.ap-northeast-1.amazonaws.com	HTTPS	216624486486	06/30/2014	Yes
Asia Pacific (Singapore)	ap-south-east-1	cloudtrail.ap-southeast-1.amazonaws.com	HTTPS	903692715234	06/30/2014	Yes

Region Name	Region	Endpoint	Protocol	AWS Account ID	Support Date	Integration with Cloud-Watch Logs
Asia Pacific (Sydney)	ap-south-east-2	cloudtrail.ap-southeast-2.amazonaws.com	HTTPS	284668455005	05/13/2014	Yes
EU (Frankfurt)	eu-central-1	cloudtrail.eu-central-1.amazonaws.com	HTTPS	035351147821	10/23/2014	Yes
EU (Ireland)	eu-west-1	cloudtrail.eu-west-1.amazonaws.com	HTTPS	859597730677	05/13/2014	Yes
South America (Sao Paulo)	sa-east-1	cloudtrail.sa-east-1.amazonaws.com	HTTPS	814480443879	06/30/2014	Coming Soon
US East (N. Virginia)	us-east-1	cloudtrail.us-east-1.amazonaws.com	HTTPS	086441151436	11/13/2013	Yes
US West (N. California)	us-west-1	cloudtrail.us-west-1.amazonaws.com	HTTPS	388731089494	05/13/2014	Yes
US West (Oregon)	us-west-2	cloudtrail.us-west-2.amazonaws.com	HTTPS	113285607260	11/13/2013	Yes

When you create a trail, you specify an Amazon S3 bucket and the region from which CloudTrail will deliver log files. Initially, CloudTrail will include only log files from that region in your bucket. You can, however, configure CloudTrail to also include log files from additional regions. For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#) (p. 72).

## CloudTrail Log File Examples

### CloudTrail Log File Name Format

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

```
AccountID_CloudTrail_RegionName_YYYYMMDDTHHmmZ_UniqueString.FileNameFormat
```

- YYYY, MM, DD, HH, and mm are the digits of the year, month, day, hour, and minute (respectively) 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 always `json.gz`, which is a JSON text file in compressed gzip format.

### Example CloudTrail Log File Name

```
111122223333_CloudTrail_ap-northeast-1_20150801T0210Z_Mu0KsOhtH1ar15ZZ.json.gz
```

## Log Examples

The following sections show selected example log entries for a few of the services that are supported by CloudTrail. A log file is made up of one or more records. The following topics do not typically display all of the records that a log file might contain. They typically show only the records for an action that started the creation of a log file.

### EC2 Log Examples

Amazon Elastic Compute Cloud (Amazon EC2) provides resizeable computing capacity in the Amazon Web Services (AWS) cloud. You can launch as many or as few virtual servers as you need, 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 about Amazon EC2, see the [Amazon EC2 User Guide](#).

The following log file record shows that an IAM user named Alice called the Amazon EC2 **StartInstances** action by using the **ec2-start-instances** CLI 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-west-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"
          }
        }]
      }
    }
  },
  ... additional entries ...
]
```

The following log file record shows that an IAM user named Alice called the Amazon EC2 **StopInstances** action by using the **ec2-stop-instances** command in the CLI.

```
{
  "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-west-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"
        }
    }
}
... additional entries ...
]
```

The following log file record shows that the Amazon EC2 console back-end called the **CreateKeyPair** action in response to requests initiated by an IAM user. Note that the **responseElements** contain a hash of the key pair and that the key material has been removed by AWS.

```
{
  "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-west-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"
      },
      ... additional entries ...
    }
  ]
}
```

## IAM Log Examples

AWS Identity and Access Management is a web service that enables AWS customers to manage users and user permissions. The service is targeted at organizations with multiple users. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access. Without IAM, organizations with multiple users and systems must either create multiple AWS accounts, each with its own billing and subscriptions to AWS products, or employees must all share the security credentials of a single AWS account. Also, without IAM, you have no control over the tasks a particular user or system can do and what AWS resources they might use. For more information, see the [IAM User Guide](#)

The following log file record shows that an IAM user called 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-1",
    "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 log file record shows that an IAM user called the **AddUserToGroup** action to add Bob to the administrators group.

```
{
  "Records": [
    {
      "eventVersion": "1.0",
      "userIdentity": {
```

## AWS CloudTrail User Guide Log Examples

```
        "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-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "AWSConsole",
    "requestParameters": {
        "userName": "Bob",
        "groupName": "admin"
    },
    "responseElements": null
},
...additional entries
]
}
```

The following log file record shows that an IAM user called the **CreateRole** action to create a new IAM role. The API was called by using a CLI command.

```
{
  "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-1",
    "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\": [\n          \"arn:aws:iam::210987654321:root\"\n        ]\n      },\n      \"Action\": \"sts:AssumeRole\"\n    }\n  ]\n}",
      "roleName": "TestRole"
    }
  ]
}
```

## AWS CloudTrail User Guide Log Examples

---

```
    },
    "responseElements": {
      "role": {
        "assumeRolePolicyDocument": "%7B%0A%20%20%22Ver
sion%22%3A%20%222012-10-17%22%2C%0A%20%20%22State
ment%22%3A%20%5B%0A%20%20%20%20%7B%0A%20%20%20%20%22Sid%22%3A%20%22%22%2C%0A%20%20%20%20%22EF
fect%22%3A%20%22Allow%22%2C%0A%20%20%20%20%20%20%22Princip
al%22%3A%20%7B%0A%20%20%20%20%20%20%20%20%20%22AWS%22%3A%20%22arn%3Aaws%3Ai
am%3A%3A803981987763%3Aroot%22%0A%20%20%20%20%20%20%7D%2C%0A%20%20%20%20%20%20%22Ac
tion%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": "/"
      }
    }
  }
}
```



# Getting Started with CloudTrail

---

CloudTrail enables you to retrieve a history of API calls and other events for your account. This includes calls and events made by the AWS Management Console and command line tools, by any of the AWS SDKs, or by other AWS services. The following topics discuss how to get started using CloudTrail.

## Topics

- [Creating and Updating Your Trail](#) (p. 21)
- [Viewing CloudTrail Events](#) (p. 30)
- [Controlling User Permissions for CloudTrail](#) (p. 63)
- [Getting and Viewing Your CloudTrail Log Files](#) (p. 67)
- [Configuring Amazon SNS Notifications for CloudTrail](#) (p. 69)

## Creating and Updating Your Trail

This section walks you through creating a trail for your AWS account. You can create trails by using the AWS CloudTrail console or the AWS command line interface (CLI). Both methods follow the same steps:

1. Turn on CloudTrail.
2. Create a new Amazon S3 bucket for storing your log files, or specify an existing bucket where you want the log files delivered.
3. (Optional) Create a new Amazon SNS topic in order to receive notifications when new log files are delivered.

In addition, the following sections explain how to update a trail and how to stop and start CloudTrail delivery of log files.

## Topics

- [Amazon S3 Bucket Naming Requirements](#) (p. 22)
- [Amazon S3 Bucket Policy for CloudTrail](#) (p. 22)
- [Creating and Updating a Trail with the CloudTrail Console](#) (p. 26)
- [Creating and Updating a Trail with the AWS CLI](#) (p. 27)

## 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:

- 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).

### Caution

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 Developer Guide.

## Amazon S3 Bucket Policy for CloudTrail

By default, all Amazon S3 buckets and objects are private. Only the resource owner, the AWS account that created the bucket, can access the bucket and any objects it contains. The resource owner can, however, choose to grant access permissions to other resources and users by writing an access policy. The following topics discuss the bucket policy necessary to enable CloudTrail to write log files to a bucket from supported regions:

### Topics

- [Creating an Amazon S3 Bucket Policy by using the CloudTrail Console or CLI \(p. 22\)](#)
- [Manually Editing the Bucket Policy for CloudTrail \(p. 23\)](#)
- [Troubleshooting Amazon S3 Bucket Policy Errors \(p. 24\)](#)

## Creating an Amazon S3 Bucket Policy by using the CloudTrail Console or CLI

When you specify an Amazon S3 bucket as the location for log file delivery, you must make sure that you attach a policy to the bucket that allows CloudTrail to work with the bucket. The easiest way to do this is to create the bucket using the CloudTrail console or the AWS CLI. When you do so, CloudTrail automatically attaches the policy to the bucket for you and fills in the following fields:

- The allowed SIDs.
- The name of the folder where the log files will be stored.
- The current and upcoming regions in which CloudTrail can operate.
- The bucket name.
- The optional prefix if you specified one at creation.
- The ID of the owning account.

## Manually Editing the Bucket Policy for CloudTrail

If you use an existing bucket, it is best to use a bucket that was created specifically for CloudTrail. Perform the following steps to attach the required policy to the bucket:

### To attach the policy required by CloudTrail to an Amazon S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket where you want CloudTrail to deliver your log files, and then click **Properties**.
3. Click **Permissions**.
4. Click **Edit Bucket Policy**.
5. Copy the following policy into the **Bucket Policy Editor** window and then substitute the correct names of your bucket, prefix, and account number for the placeholders indicated in italics. If you specified a prefix when you created your trail, be sure to include it here. The prefix is an optional addition to the Amazon S3 object key that helps create a folder-like organization in your bucket.

The following policy enables CloudTrail to write log files to your bucket from any of the currently [CloudTrail Supported Regions \(p. 13\)](#).

#### Caution

If the existing bucket already has one or more policies attached to it, add the statements for CloudTrail access to that policy or policies. We recommend that you evaluate the resulting set of permissions to be sure that they are appropriate for the users who will be accessing the bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck20131101",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::903692715234: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",
          "arn:aws:iam::035351147821:root"
        ]
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::myBucketName"
    },
    {
      "Sid": "AWSCloudTrailWrite20131101",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::903692715234: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",
        "arn:aws:iam::035351147821:root"
    ]
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3::myBucketName/[optional] prefix/AWSLogs/myAccountID/*",
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}
]
```

### Important

If you are using non-isolated regions, do not add the CloudTrail AWS Account ID of an isolated region like AWS GovCloud (US) to your policy templates, or an 'Invalid principal in policy' error will occur. Similarly, if you are using an isolated region like AWS GovCloud (US), adding the CloudTrail AWS Account ID of non-isolated regions to your policy templates will also produce this error. For more information, see the [CloudTrail topic](#) in the AWS GovCloud (US) User Guide.

Typically, when you create a trail, you specify the region that CloudTrail will deliver log files from. However, CloudTrail can also aggregate log files from other regions and accounts into your Amazon S3 bucket as long as those regions have write access to your bucket. For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#) (p. 72). For more information about the regions supported by CloudTrail, see [CloudTrail Supported Regions](#) (p. 13). For more information about using the Amazon S3 console to create or edit bucket policies, see the topic [Editing Bucket Permissions](#) in the *Amazon Simple Storage Service Console User Guide*.

## Troubleshooting Amazon S3 Bucket Policy Errors

### When specifying an existing bucket for a new region

If you decide to use an existing bucket when you turn on CloudTrail for a new region, you might receive the error **There is a problem with the bucket policy**. If so, it is possible that your bucket policy does not enable access for the new region. For example, you might receive this error if your bucket policy supports only the us-east-1 (US East (N. Virginia)) and us-west-2 (US West (Oregon)) regions and you try to turn on your trail in ap-southeast-2 (Asia Pacific (Sydney)). For more information about the regions currently supported by CloudTrail, see [CloudTrail Supported Regions](#) (p. 13).

To resolve this issue, do one of the following:

- Use the CloudTrail console or CLI to turn on CloudTrail and specify a new bucket. CloudTrail automatically attaches the correct policy to the bucket for you.
- For an existing bucket, follow the preceding steps in the [Manually Editing the Bucket Policy for CloudTrail](#) (p. 23) section by copying the listed policy straight into your Amazon S3 policy editor, remembering to substitute the correct names of your bucket, prefix, and account number for the placeholders indicated in italics.
- Also for an existing bucket, you can manually enter in the Amazon S3 bucket policy only those regions that you want to support. For example, if your bucket already supports the us-east-1 (US East (N. Virginia)) and us-west-2 (US West (Oregon)) regions and you want to add support for us-west-1 (US

West (N. California)) and ap-southeast-2 (Asia Pacific (Sydney)), edit the **Principal** sections of the Amazon S3 bucket policy to include the following ARNs.

```
...
  "Principal": {
    "AWS": [
      "arn:aws:iam::086441151436:root",
      "arn:aws:iam::113285607260:root",
      "arn:aws:iam::388731089494:root",
      "arn:aws:iam::284668455005:root"
    ]
  },
  ...
```

### When adding, changing, or removing a prefix for an existing bucket

If you try to add, modify, or remove a log file prefix for the existing Amazon S3 bucket of a CloudTrail configuration, you might receive the error **There is a problem with the bucket policy**. To resolve this issue, first use the Amazon S3 console to update the prefix in the bucket's policy, and then use the CloudTrail console to specify the same prefix for the bucket in CloudTrail configuration.

#### To update the log file prefix for an Amazon S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket for which you want to modify the prefix, and then click **Properties**.
3. Click **Permissions**.
4. Click **Edit Bucket Policy**.
5. In the bucket policy, under the `s3:PutObject` action, edit the `Resource` entry to add, modify, or remove the log file *prefix* as required.

```
"Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::myBucketName/prefix/AWSLogs/myAccountID/*",
```

6. Click **Save**.
7. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
8. On the **Configuration** page, click the pencil icon in upper right corner of the **S3** box to edit the settings for your S3 bucket.
9. Click **Advanced**.
10. For **S3 bucket**, choose the bucket whose prefix you are changing.
11. In the **Log file prefix** box, update the prefix to match the prefix that you entered in the S3 bucket policy.
12. Click **Save**.

# Creating and Updating a Trail with the CloudTrail Console

## Creating a Trail

The following steps create a CloudTrail trail, including an Amazon S3 bucket and an optional Amazon SNS topic to which you can subscribe for notifications that new log files are available. While you can use an existing bucket, we recommend that you create a new one. When you create a new bucket, CloudTrail creates the necessary IAM policies for you on the bucket and topic.

Any trail you create using the console has the name "Default." If you want to specify a name for your trail, use the `update-subscription` command from the CLI, which is described in the next section. Currently, you can create one trail for each region where the service is supported. If you want to replace a trail and create a new one, first use the `delete-trail` command from the CLI.

### To create a CloudTrail trail using the AWS Management Console

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/home>.
2. Click **Get Started Now**.
3. On the **Turn On CloudTrail** page, next to **Create a new S3 bucket?**, select **Yes** to create a new bucket or **No** to use an existing one.

#### Note

If you click **No**, the console options change to provide a drop-down list from which you can select an existing Amazon S3 bucket. If you use an existing bucket, you must manually edit the bucket policy to grant CloudTrail permission to write to it. You can only designate an existing bucket owned by the account under which the trail is created. See the section [Amazon S3 Bucket Policy for CloudTrail \(p. 22\)](#) for information about manually editing the policy for a bucket.

4. In the **S3 bucket name** field, enter a name for the bucket you want to designate for log file storage. The name that you choose must be globally unique. For more information about S3 bucket naming rules and conventions, see [Amazon S3 Bucket Naming Requirements \(p. 22\)](#).
5. (Optional) If you want to enter a prefix for your bucket, subscribe to global services such as IAM or AWS STS, or create an Amazon SNS topic, click **Advanced**.
6. (Optional) In the **Log file prefix** field, enter a prefix for your Amazon S3 bucket. The prefix is an addition to the URL for an Amazon S3 object that helps create a folder-like organization in your bucket. The location where your log files will be stored is shown under the text field.
7. (Optional) Select **Yes** or **No** for **SNS notification for every log file delivery?** If you select **Yes**, enter a name for your Amazon SNS topic in the **SNS topic (new)** field.

#### Note

If you create a topic, you must subscribe to it in order to get notification of log file delivery. Due to the frequency of notifications, we recommend that you configure the subscription to use an Amazon SQS queue to handle notifications programmatically. See the [Amazon Simple Notification Service Getting Started Guide](#) for information.

#### Note

CloudTrail stores multiple events in a log file. SNS notifications are per file, not per event.

8. Click **Turn On**.
9. In about 15 minutes, CloudTrail starts publishing log files that show the AWS API calls made in your account since you completed the preceding steps.

## Updating a Trail

### To update a trail using the AWS Management Console

1. Navigate to the **Configuration** page of the CloudTrail console.
2. To edit the settings for your S3 bucket, click the pencil icon in upper right corner of the **S3** box.
3. To optionally configure CloudTrail to deliver events to CloudWatch Logs for real-time monitoring, click **Configure** in the **CloudWatch Logs (Optional)** box. For more information, see [Sending CloudTrail Events to CloudWatch Logs](#) (p. 92).
4. (Optional) To include or exclude global services, click the pencil icon to the right of **Additional Configuration** to edit the **Include global services** option. When enabled, this option records API calls from global services such as IAM or AWS STS. In most circumstances, you should accept the default setting of **Yes**. To learn about advanced scenarios in which it is appropriate to select **No**, see the topic [Receiving CloudTrail Log Files from Multiple Sources in a Single Amazon S3 Bucket](#) (p. 72).

#### Note

We recommend including global services. If you are aggregating trails from multiple regions into one bucket, we recommend that you include global services in only one trail. This avoids generating duplicate entries for the global events in the log files, which would otherwise occur because global services generate the same events in all regions.

5. Click **Save**.

#### Note

When you designate a different bucket for your trail, we recommend that you use the same bucket name prefix that you used previously. Otherwise, you must manually update the bucket policy with the changed prefix.

## Creating and Updating a Trail with the AWS CLI

You can create a trail using the `create-subscription` command. The command uses the following options to specify the additional settings for the trail:

- `--name` specifies the name of the trail. The name 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`)
- `--s3-use-bucket` specifies an existing Amazon S3 bucket for log file storage.
- `--s3-new-bucket` specifies the name of the new bucket created when the command executes. The name of the bucket must be globally unique. For more information, see [Amazon S3 Bucket Naming Requirements](#) (p. 22).
- `--s3-prefix` specifies a prefix for the log file delivery path (optional). The maximum length is 200 characters.

#### Note

If you want to use a new log file prefix for an existing bucket, add the prefix to the bucket policy first. For more information, see [When adding, changing, or removing a prefix for an existing bucket](#) (p. 25)

- `--sns-new-topic` specifies the name of the Amazon SNS topic to which you can subscribe for notification of log file delivery to your bucket (optional).

As part of defining the CloudTrail trail, the `create-subscription` command can create a new Amazon S3 bucket for log file delivery and a new Amazon SNS topic for notifications. In contrast to trails created using the console, every trail created with the AWS CLI must have a name. You specify this name with command line operations. Again, you are limited to one trail per account for each region in which the account is running AWS resources.

**Note**

The AWS CLI commands shown below require that you have the AWS command line tools. For more information, see the [AWS Command Line Interface User Guide](#).

The following example command demonstrates the creation of a trail for an account using the AWS CLI.

```
aws cloudtrail create-subscription --name=awscloudtrail-example --s3-new-bucket=awscloudtrail-new-bucket-example --s3-prefix=prefix-example --sns-new-topic=awscloudtrail-example-log-deliverytopic
```

If the command executes successfully, you see output similar to the following:

```
CloudTrail configuration:
{
  "trailList": [
    {
      "S3KeyPrefix": "prefix-example",
      "IncludeGlobalServiceEvents": true,
      "Name": "awscloudtrail-example",
      "SnsTopicName": "awscloudtrail-example-log-deliverytopic",
      "S3BucketName": "awscloudtrail-new-bucket-example"
    }
  ]
}
```

You can update your trail using the command `update-subscription` and setting the options to new values. The following example designates a different Amazon S3 bucket. If you want a trail with a different name, you can delete the trail and run `create-subscription` again.

```
aws cloudtrail update-subscription --name=awscloudtrail-example --s3-use-bucket=awscloudtrail-new-bucket-example2 --s3-prefix=prefix-example
```

**Note**

Trails that are created by using the CloudTrail console automatically receive the name of **Default**.

If the command executes successfully, you see that the `trailList` structure has been updated:

```
CloudTrail configuration:
{
  "trailList": [
    {
      "S3KeyPrefix": "prefix-example",
      "IncludeGlobalServiceEvents": true,
      "Name": "awscloudtrail-example",
      "SnsTopicName": "awscloudtrail-example-log-deliverytopic",
```



```
    "S3BucketName": "awscloudtrail-new-bucket-example2"
  }
]
}
```

**Tip**

If you designate an existing bucket for log file publication, we recommend that you use one that was created with the CloudTrail console or CLI, as it will already have the necessary policy applied. If you decide to use an existing bucket that was not created with CloudTrail, see the topic [Amazon S3 Bucket Policy for CloudTrail](#) (p. 22) for a procedure that shows you how to apply the necessary policy.

## Additional AWS CLI Commands

The CloudTrail CLI includes several other commands that help you manage your trails. This section demonstrates how to use these commands.

### Retrieving Trail Settings and the Status of a Trail

Use the following command to retrieve trail settings:

```
aws cloudtrail describe-trails
```

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

```
{
  "trailList": [
    {
      "S3KeyPrefix": "prefix-example",
      "IncludeGlobalServiceEvents": true,
      "Name": "awscloudtrail-example",
      "SnsTopicName": "awscloudtrail-example-log-deliverytopic",
      "S3BucketName": "awscloudtrail-new-bucket-example2"
    }
  ]
}
```

Use the following command to retrieve the status of a trail. Trails that were created by using the CloudTrail console have the name of "Default".

```
aws cloudtrail get-trail-status --name awscloudtrail-example
```

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

```
{
  "LatestDeliveryAttemptTime": "2013-11-12T20:18:27Z",
  "LatestNotificationAttemptSucceeded": "2013-11-12T20:18:27Z",
  "LatestDeliveryAttemptSucceeded": "2013-11-12T20:18:27Z",
  "IsLogging": true,
  "TimeLoggingStarted": "2013-11-12T20:19:30Z",
  "LatestNotificationAttemptTime": "2013-11-12T20:18:27Z",
  "TimeLoggingStopped": "2013-11-12T20:19:16Z"
}
```

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.

## Stopping and Starting Logging for a Trail

The following commands start and stop CloudTrail logging, respectively. Trails that were created by using the CloudTrail console have the name of "Default".

```
aws cloudtrail start-logging --name awscloudtrail-example
```

```
aws cloudtrail stop-logging --name awscloudtrail-example
```

### Caution

Before deleting a bucket, you should use `stop-logging` to end all logging to the bucket. If you don't stop logging, CloudTrail will continue to attempt to deliver log files to a bucket of the same name for a limited period of time.

## Deleting a Trail

You can delete a trail using the following command. Trails that were created by using the CloudTrail console have the name of "Default".

```
aws cloudtrail delete-trail --name awscloudtrail-example
```

When you delete a trail, you do not delete either the Amazon S3 bucket or the Amazon SNS topic associated with it. If you want to delete these items, do so separately using the AWS Management Console, AWS CLI, or service API.

# Viewing CloudTrail Events

You can look up API activity captured by CloudTrail (that is, CloudTrail events) from within the CloudTrail console to troubleshoot operational and security incidents quickly. You can look up events related to the creation, modification, or deletion of resources in your AWS account. You can also filter events to make your results more targeted. Events can be looked up by using the AWS CloudTrail console, or programmatically by using the AWS SDKs or AWS Command Line Interface. This section describes how to look up events by using the CloudTrail console and the AWS CLI. For information on using the `LookupEvents` API to retrieve information from CloudTrail events, see the [AWS CloudTrail API Reference](#).

### Note

When you stop logging, CloudTrail stops delivering events to your Amazon S3 bucket. As a result, any events that occurred when CloudTrail logging was disabled will not be captured, and they will not be available for viewing.

You can use the CloudTrail console, AWS CLI, or AWS SDKs to review API activity for a region for the times in which you had CloudTrail turned on in that region during the last seven days. For information on other ways to get and view CloudTrail log files, including those older than seven days, see [Getting and Viewing Your CloudTrail Log Files](#) (p. 67).

## Viewing CloudTrail Events in the CloudTrail Console

After you turn on logging with CloudTrail, you can view the API activity that occurred in your AWS account during the last seven days from the CloudTrail console. For a list of supported services, see [Services Supported for Event Lookup in CloudTrail \(p. 38\)](#). For a list of supported regions, see [Regions Supported for Event Lookup in CloudTrail \(p. 37\)](#).

### To view CloudTrail events

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/home/>.
2. In the navigation pane, choose **API Activity History**.

A list of events, with the latest event first, appears in the content pane. To see more events, scroll down. If no events have been logged, none will be displayed.

## Filtering CloudTrail Events

You can filter events by the following five optional attributes. Currently, you can filter by time range plus a maximum of one other attribute at a time.

- User name
- Event name
- Resource type
- Resource name
- Time range

For a list of the resource types supported for event lookup, see [Resource Types Supported for Event Lookup in CloudTrail \(p. 60\)](#)

### To filter by attribute

1. To filter the results by an attribute, click **Select attribute**, and then enter or choose a value in the **Enter lookup value** box on the right.
2. To remove an attribute filter, click the **X** on the right of the attribute filter box.

### To filter by a start and end date and time

1. To narrow the time range for the events that you want to see, click **Select time range**.
2. To remove an time range filter, click the calendar icon on the right of the date range box, and then choose **Remove**.

If there are no events logged for the attribute and/or time that you have selected, the results list is empty.

#### Note

You can apply only one attribute filter in addition to the time range at a time. Choosing a second attribute filter replaces the first attribute filter while preserving your specified time range.

## Viewing Details for an Event

1. Click an event in the result list to show its details. You can open and close events independently.

2. If the event referenced more than one resource, the additional resources are listed at the bottom of the details pane.
3. Some referenced resources have links. Click a referenced resource link to open the console for the resource.
4. To view the event in JSON format, click **View Event** in the details pane.
5. To close a details pane for an event, click the event in the results list again.

## Viewing CloudTrail Events with the AWS CLI

You can look up CloudTrail events for the last seven days using the `aws cloudtrail lookup-events` command. `lookup-events` has the following options:

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

These options are explained in this topic. For a list of services supported for event lookup, see [Services Supported for Event Lookup in CloudTrail \(p. 38\)](#). For a list of regions supported for event lookup, see [Regions Supported for Event Lookup in CloudTrail \(p. 37\)](#).

For general information on using the AWS Command Line Interface, see the [AWS Command Line Interface User Guide](#).

### Prerequisites

- To run AWS CLI commands, you must first install the AWS CLI. For information, see [Installing the AWS Command Line Interface](#).
- Make sure your AWS 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 AWS CLI session, use the `aws configure` command. For more information, see [Configuring the AWS Command Line Interface](#).

#### Note

The CloudTrail AWS 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

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

```
aws cloudtrail lookup-events
```

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

```
{
  "NextToken": "kbOt5LlZe++mErCebpy2TgaMgmDvF1kYG
FcH64JSjIbZfjsuvrSgg66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXdl2juy3CIZW8=",
  "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-1\",
        \"sourceIPAddress\": \"203.0.113.4\",
        \"userAgent\": \"Mozilla/5.0\",
        \"requestParameters\": null,
        \"responseElements\": {\"ConsoleLogin\": \"Success\"},
        \"additionalEventData\": {
          \"MobileVersion\": \"No\",
          \"LoginTo\": \"https://console.aws.amazon.com/con
sole/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 \(p. 37\)](#) later in this document. For an explanation of the fields in the CloudTrail event, see [CloudTrail Record Contents \(p. 144\)](#).

## Specifying the number of events to return

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

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

The default value for *<integer>* is 10. Possible values are 1 through 50. The following example returns one result.

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

## Looking up events by time range

Events from the past seven 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 seven 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>
```

### Note

Although `--lookup-attributes` is technically a list, currently you can specify only one attribute key/value pair per lookup.

Possible values for `AttributeKey` are the following. These names are case sensitive.

- EventId
- EventName
- Username
- ResourceType
- ResourceName

For a list of the resource types supported for event lookup, see [Resource Types Supported for Event Lookup in CloudTrail \(p. 60\)](#)

## Attribute lookup examples

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 `Username` is `root`.

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

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 `ResourceName` is `CloudTrail_CloudWatchLogs_Role`.

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

## 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++mErCebpy2TgaMgmDvF1kYG  
FcH64JSjIbZfjsuvrSqq66b5YGssKutDYIyII4lrP4IDbeQdiObkp9YAlju3oXd12juy3CIZW8=
```

## Getting JSON input from a file

The AWS CLI for some AWS 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

**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.

**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.

**EventId**

A string that contains the GUID of the event returned.

**Username**

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

**EventTime**

The date and time, in UNIX time format, of the 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](#).

**EventName**

A string that contains the name of the event returned.

**Resources**

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

**ResourceType**

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

**ResourceName**

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

## Regions Supported for Event Lookup in CloudTrail

You can lookup CloudTrail events for the supported services in the following regions:

Region Name	Region
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
South America (Sao Paulo)	sa-east-1
US East (N. Virginia)	us-east-1

Region Name	Region
US West (N. California)	us-west-1
US West (Oregon)	us-west-2

## Services Supported for Event Lookup in CloudTrail

You can look up CloudTrail events in the supported regions for the following services:

### **Analytics**

- AWS Data Pipeline
- Amazon Elastic MapReduce
- Amazon Kinesis
- Amazon Redshift

### **Application Services**

- Amazon CloudSearch
- Amazon Elastic Transcoder
- Amazon Simple Workflow Service

### **Compute**

- Auto Scaling
- AWS Elastic Beanstalk
- Amazon EC2
- Elastic Load Balancing

### **Database**

- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Relational Database Service

### **Developer Tools**

- AWS CodeDeploy

### **Enterprise Applications**

- Amazon WorkDocs

### **Management Tools**

- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch

- [AWS Config](#)
- [AWS OpsWorks](#)

### **Networking**

- [AWS Direct Connect](#)
- [Amazon Route 53](#)

### **Security and Identity**

- [AWS CloudHSM](#)
- [AWS Identity and Access Management](#)
- [AWS Management Console Sign-in Service](#)

### **Storage and Content Delivery**

- [Amazon CloudFront](#)
- [Amazon Glacier](#)
- [Amazon S3 bucket level events](#)
- [AWS Storage Gateway](#)

## **APIs Supported for Event Lookup in CloudTrail**

This page lists the API calls supported for event lookup in the CloudTrail API Activity History feature. The APIs are listed by service.

### **Topics**

- [Auto Scaling APIs \(p. 40\)](#)
- [AWS CloudFormation APIs \(p. 41\)](#)
- [Amazon CloudFront APIs \(p. 41\)](#)
- [AWS CloudHSM APIs \(p. 41\)](#)
- [Amazon CloudSearch APIs \(p. 42\)](#)
- [AWS CloudTrail APIs \(p. 43\)](#)
- [Amazon CloudWatch APIs \(p. 43\)](#)
- [AWS CodeDeploy APIs \(p. 43\)](#)
- [AWS Config APIs \(p. 43\)](#)
- [AWS Data Pipeline APIs \(p. 44\)](#)
- [AWS Direct Connect APIs \(p. 44\)](#)
- [Amazon DynamoDB APIs \(p. 44\)](#)
- [Amazon EC2 APIs \(p. 45\)](#)
- [Amazon ElastiCache APIs \(p. 48\)](#)
- [AWS Elastic Beanstalk APIs \(p. 49\)](#)
- [Elastic Load Balancing APIs \(p. 50\)](#)
- [Amazon Elastic MapReduce APIs \(p. 51\)](#)
- [Amazon Elastic Transcoder APIs \(p. 51\)](#)
- [Amazon Glacier APIs \(p. 51\)](#)
- [AWS Identity and Access Management APIs \(p. 52\)](#)

- [Amazon Kinesis APIs \(p. 54\)](#)
- [AWS OpsWorks APIs \(p. 54\)](#)
- [Amazon RDS APIs \(p. 55\)](#)
- [Amazon Redshift APIs \(p. 57\)](#)
- [Amazon Route 53 APIs \(p. 58\)](#)
- [Amazon S3 Bucket Level APIs \(p. 58\)](#)
- [AWS Storage Gateway APIs \(p. 59\)](#)
- [Amazon Simple Workflow Service APIs \(p. 60\)](#)
- [Amazon WorkDocs APIs \(p. 60\)](#)

## Auto Scaling APIs

AttachInstances
CompleteLifecycleAction
CreateAutoScalingGroup
CreateLaunchConfiguration
CreateOrUpdateTags
DeleteAutoScalingGroup
DeleteLaunchConfiguration
DeleteLifecycleHook
DeleteNotificationConfiguration
DeletePolicy
DeleteScheduledAction
DeleteTags
DetachInstances
DisableMetricsCollection
EnableMetricsCollection
EnterStandby
ExecutePolicy
ExitStandby
PutLifecycleHook
PutNotificationConfiguration
PutScalingPolicy
PutScheduledUpdateGroupAction
RecordLifecycleActionHeartbeat
ResumeProcesses

SetDesiredCapacity
SetInstanceHealth
SuspendProcesses
TerminateInstanceInAutoScalingGroup
UpdateAutoScalingGroup

## **AWS CloudFormation APIs**

CancelUpdateStack
CreateStack
DeleteStack
SetStackPolicy
SignalResource
UpdateStack

## **Amazon CloudFront APIs**

CreateCloudFrontOriginAccessIdentity
CreateDistribution
CreateInvalidation
CreateStreamingDistribution
DeleteCloudFrontOriginAccessIdentity
DeleteDistribution
DeleteStreamingDistribution
UpdateCloudFrontOriginAccessIdentity
UpdateDistribution
UpdateStreamingDistribution

## **AWS CloudHSM APIs**

AdminCreateHsm
CreateHapg
CreateHsm
CreateLunaClient

DeleteHapg
DeleteHsm
DeleteLunaClient
ModifyHapg
ModifyHsm
ModifyLunaClient

## Amazon CloudSearch APIs

BuildSuggesters
CreateDomain
DefineAnalysisScheme
DefineExpression
DefineIndexField
DefineIndexFields
DefineRankExpression
DefineSuggester
DeleteAnalysisScheme
DeleteDomain
DeleteExpression
DeleteIndexField
DeleteRankExpression
DeleteSuggester
IndexDocuments
UpdateAvailabilityOptions
UpdateDefaultSearchField
UpdateScalingParameters
UpdateServiceAccessPolicies
UpdateStemmingOptions
UpdateStopwordOptions
UpdateSynonymOptions

## AWS CloudTrail APIs

CreateTrail
DeleteTrail
StartLogging
StopLogging
UpdateTrail

## Amazon CloudWatch APIs

DeleteAlarms
DisableAlarmActions
EnableAlarmActions
PutMetricAlarm
SetAlarmState

## AWS CodeDeploy APIs

CreateApplication
CreateDeployment
CreateDeploymentConfig
CreateDeploymentGroup
DeleteApplication
DeleteDeploymentConfig
DeleteDeploymentGroup
RegisterApplicationRevision
StopDeployment
UpdateApplication
UpdateDeploymentGroup

## AWS Config APIs

DeleteDeliveryChannel
PutConfigurationRecorder
PutDeliveryChannel

StartConfigurationRecorder
----------------------------

StopConfigurationRecorder
---------------------------

## **AWS Data Pipeline APIs**

ActivatePipeline
------------------

CreatePipeline
----------------

DeletePipeline
----------------

PutPipelineDefinition
-----------------------

SetStatus
-----------

## **AWS Direct Connect APIs**

AllocateConnectionOnInterconnect
----------------------------------

AllocatePrivateVirtualInterface
---------------------------------

AllocatePublicVirtualInterface
--------------------------------

ConfirmConnection
-------------------

ConfirmPrivateVirtualInterface
--------------------------------

ConfirmPublicVirtualInterface
-------------------------------

CreateConnection
------------------

CreateInterconnect
--------------------

CreatePrivateVirtualInterface
-------------------------------

CreatePublicVirtualInterface
------------------------------

DeleteConnection
------------------

DeleteInterconnect
--------------------

DeleteVirtualInterface
------------------------

## **Amazon DynamoDB APIs**

CreateTable
-------------

DeleteTable
-------------

PurchaseReservedCapacityOfferings
-----------------------------------

UpdateTable
-------------



## Amazon EC2 APIs

AcceptVpcPeeringConnection
AllocateAddress
AssignPrivateIpAddresses
AssociateAddress
AssociateDhcpOptions
AssociateRouteTable
AttachClassicLinkVpc
AttachInternetGateway
AttachNetworkInterface
AttachVolume
AttachVpnGateway
AuthorizeSecurityGroupEgress
AuthorizeSecurityGroupIngress
BundleInstance
CancelBundleTask
CancelConversionTask
CancelExportTask
CancelImportTask
CancelReservedInstancesListing
CancelScheduledInstanceRequest
CancelSpotInstanceRequests
CopyImage
CopySnapshot
CreateCustomerGateway
CreateDhcpOptions
CreateImage
CreateInstance
CreateInstanceExportTask
CreateInternetGateway
CreateKeyPair
CreateNetworkAcl

**AWS CloudTrail User Guide**  
**APIs Supported for Event Lookup in CloudTrail**

---

CreateNetworkAclEntry
CreateNetworkInterface
CreatePlacementGroup
CreateReservedInstancesListing
CreateRoute
CreateRouteTable
CreateSecurityGroup
CreateSnapshot
CreateSpotDatafeedSubscription
CreateSubnet
CreateTags
CreateVolume
CreateVpc
CreateVpcEndpoint
CreateVpcPeeringConnection
CreateVpnConnection
CreateVpnConnectionRoute
CreateVpnGateway
DeleteCustomerGateway
DeleteDhcpOptions
DeleteInternetGateway
DeleteKeyPair
DeleteNetworkAcl
DeleteNetworkAclEntry
DeleteNetworkInterface
DeletePlacementGroup
DeleteRoute
DeleteRouteTable
DeleteSecurityGroup
DeleteSnapshot
DeleteSpotDatafeedSubscription
DeleteSubnet
DeleteTags

**AWS CloudTrail User Guide**  
**APIs Supported for Event Lookup in CloudTrail**

---

DeleteVolume
DeleteVpc
DeleteVpcEndpoints
DeleteVpcPeeringConnection
DeleteVpnConnection
DeleteVpnConnectionRoute
DeleteVpnGateway
DeregisterImage
DetachClassicLinkVpc
DetachInternetGateway
DetachNetworkInterface
DetachVolume
DetachVpnGateway
DisableVgwRoutePropagation
DisableVpcClassicLink
DisassociateAddress
DisassociateRouteTable
EnableVgwRoutePropagation
EnableVolumeIO
EnableVpcClassicLink
ImportImage
ImportInstance
ImportKeyPair
ImportSnapshot
ImportVolume
ModifyImageAttribute
ModifyInstanceAttribute
ModifyNetworkInterfaceAttribute
ModifyReservedInstances
ModifySnapshotAttribute
ModifySubnetAttribute
ModifyVolumeAttribute
ModifyVpcAttribute

ModifyVpcEndpoint
MonitorInstances
MoveAddressToVpc
PurchaseReservedInstancesOffering
RebootInstances
RegisterImage
RejectVpcPeeringConnection
ReleaseAddress
ReplaceNetworkAclAssociation
ReplaceNetworkAclEntry
ReplaceRoute
ReplaceRouteTableAssociation
RequestScheduledInstances
RequestSpotInstances
ResetImageAttribute
ResetInstanceAttribute
ResetNetworkInterfaceAttribute
ResetSnapshotAttribute
RestoreAddressToClassic
RevokeSecurityGroupEgress
RevokeSecurityGroupIngress
RunInstances
StartInstances
StopInstances
TerminateInstances
UnassignPrivateIpAddresses
UnmonitorInstances

## **Amazon ElastiCache APIs**

AuthorizeCacheSecurityGroupIngress
CopySnapshot
CreateCacheCluster

CreateCacheParameterGroup
CreateCacheSecurityGroup
CreateCacheSubnetGroup
CreateReplicationGroup
CreateSnapshot
DeleteCacheCluster
DeleteCacheParameterGroup
DeleteCacheSecurityGroup
DeleteCacheSubnetGroup
DeleteReplicationGroup
DeleteSnapshot
ModifyCacheCluster
ModifyCacheParameterGroup
ModifyCacheSubnetGroup
ModifyReplicationGroup
PurchaseReservedCacheNodesOffering
RebootCacheCluster
ResetCacheParameterGroup
RevokeCacheSecurityGroupIngress

## **AWS Elastic Beanstalk APIs**

AbortEnvironmentUpdate
CreateApplication
CreateApplicationVersion
CreateConfigurationTemplate
CreateEnvironment
CreateStorageLocation
DeleteApplication
DeleteApplicationVersion
DeleteConfigurationTemplate
DeleteEnvironmentConfiguration
RebuildEnvironment

RestartAppServer
SwapEnvironmentCNAMEs
TerminateEnvironment
UpdateApplication
UpdateApplicationVersion
UpdateConfigurationTemplate
UpdateEnvironment

## Elastic Load Balancing APIs

AddTags
ApplySecurityGroupsToLoadBalancer
AttachLoadBalancerToSubnets
ConfigureHealthCheck
CreateAppCookieStickinessPolicy
CreateLBCookieStickinessPolicy
CreateLoadBalancer
CreateLoadBalancerListeners
CreateLoadBalancerPolicy
DeleteLoadBalancer
DeleteLoadBalancerListeners
DeleteLoadBalancerPolicy
DeregisterInstancesFromLoadBalancer
DetachLoadBalancerFromSubnets
DisableAvailabilityZonesForLoadBalancer
EnableAvailabilityZonesForLoadBalancer
ModifyLoadBalancerAttributes
RegisterInstancesWithLoadBalancer
RemoveTags
SetLoadBalancerListenerSSLCertificate
SetLoadBalancerPoliciesForBackendServer
SetLoadBalancerPoliciesOfListener

## Amazon Elastic MapReduce APIs

AddInstanceGroups
AddJobFlowSteps
AddTags
ModifyInstanceGroups
RemoveTags
RunJobFlow
SetTerminationProtection
SetVisibleToAllUsers
TerminateJobFlows

## Amazon Elastic Transcoder APIs

CancelJob
CreateJob
CreatePipeline
CreatePreset
DeletePipeline
DeletePreset
TestRole
UpdatePipeline
UpdatePipelineNotifications
UpdatePipelineStatus

## Amazon Glacier APIs

CreateVault
DeleteVault
DeleteVaultNotifications
SetDataRetrievalPolicy
SetVaultNotifications

## AWS Identity and Access Management APIs

AddClientIDToOpenIDConnectProvider
AddRoleToInstanceProfile
AddUserToGroup
AttachGroupPolicy
AttachRolePolicy
AttachUserPolicy
ChangePassword
CreateAccessKey
CreateAccountAlias
CreateGroup
CreateInstanceProfile
CreateLoginProfile
CreateOpenIDConnectProvider
CreatePolicy
CreatePolicyVersion
CreateRole
CreateSAMLProvider
CreateUser
CreateVirtualMFADevice
DeactivateMFADevice
DeleteAccessKey
DeleteAccountAlias
DeleteAccountPasswordPolicy
DeleteGroup
DeleteGroupPolicy
DeleteInstanceProfile
DeleteLoginProfile
DeleteOpenIDConnectProvider
DeletePolicy
DeletePolicyVersion
DeleteRole



**AWS CloudTrail User Guide**  
**APIs Supported for Event Lookup in CloudTrail**

---

DeleteRolePolicy
DeleteSAMLProvider
DeleteServerCertificate
DeleteSigningCertificate
DeleteUser
DeleteUserPolicy
DeleteVirtualMFADevice
DetachGroupPolicy
DetachRolePolicy
DetachUserPolicy
EnableMFADevice
PutGroupPolicy
PutRolePolicy
PutUserPolicy
RemoveClientIDFromOpenIDConnectProvider
RemoveRoleFromInstanceProfile
RemoveUserFromGroup
ResyncMFADevice
SetDefaultPolicyVersion
UpdateAccessKey
UpdateAccountPasswordPolicy
UpdateAssumeRolePolicy
UpdateGroup
UpdateLoginProfile
UpdateOpenIDConnectProviderThumbprint
UpdateSAMLProvider
UpdateServerCertificate
UpdateSigningCertificate
UpdateUser
UploadServerCertificate
UploadSigningCertificate

## Amazon Kinesis APIs

CreateStream
DeleteStream
MergeShards
SplitShard

## AWS OpsWorks APIs

AssignInstance
AssignVolume
AssociateElasticIp
AttachElasticLoadBalancer
CloneStack
CreateApp
CreateDeployment
CreateInstance
CreateLayer
CreateStack
CreateUserProfile
DeleteApp
DeleteInstance
DeleteLayer
DeleteStack
DeleteUserProfile
DeregisterElasticIp
DeregisterInstance
DeregisterRdsDbInstance
DeregisterVolume
DetachElasticLoadBalancer
DisassociateElasticIp
RebootInstance
RegisterElasticIp
RegisterInstance

RegisterRdsDbInstance
RegisterVolume
SetLoadBasedAutoScaling
SetPermission
SetTimeBasedAutoScaling
StartInstance
StartStack
StopInstance
StopStack
UnassignInstance
UnassignVolume
UpdateApp
UpdateElasticIp
UpdateInstance
UpdateLayer
UpdateMyUserProfile
UpdateRdsDbInstance
UpdateStack
UpdateUserProfile
UpdateVolume

## Amazon RDS APIs

AddSourceIdentifierToSubscription
AddTagsToResource
ApplyPendingMaintenanceAction
AuthorizeDBSecurityGroupIngress
CopyDBParameterGroup
CopyDBSnapshot
CopyOptionGroup
CreateDBCluster
CreateDBClusterSnapshot
CreateDBInstance

**AWS CloudTrail User Guide**  
**APIs Supported for Event Lookup in CloudTrail**

---

CreateDBInstanceReadReplica
CreateDBParameterGroup
CreateDBSecurityGroup
CreateDBSnapshot
CreateDBSubnetGroup
CreateEventSubscription
CreateOptionGroup
DeleteDBCluster
DeleteDBClusterSnapshot
DeleteDBInstance
DeleteDBParameterGroup
DeleteDBSecurityGroup
DeleteDBSnapshot
DeleteDBSubnetGroup
DeleteEventSubscription
DeleteOptionGroup
FailoverDBCluster
ModifyDBCluster
ModifyDBInstance
ModifyDBParameterGroup
ModifyDBSubnetGroup
ModifyEventSubscription
ModifyOptionGroup
PromoteReadReplica
PurchaseReservedDBInstancesOffering
RebootDBInstance
RemoveSourceIdentifierFromSubscription
RemoveTagsFromResource
ResetDBParameterGroup
RestoreDBClusterFromSnapshot
RestoreDBInstanceFromDBSnapshot
RestoreDBInstanceToPointInTime
RevokeDBSecurityGroupIngress

## Amazon Redshift APIs

AuthorizeClusterSecurityGroupIngress
AuthorizeSnapshotAccess
CopyClusterSnapshot
CreateCluster
CreateClusterParameterGroup
CreateClusterSecurityGroup
CreateClusterSnapshot
CreateClusterSubnetGroup
CreateEventSubscription
CreateHsmClientCertificate
CreateHsmConfiguration
CreateTags
DeleteCluster
DeleteClusterParameterGroup
DeleteClusterSecurityGroup
DeleteClusterSnapshot
DeleteClusterSubnetGroup
DeleteEventSubscription
DeleteHsmClientCertificate
DeleteHsmConfiguration
DeleteTags
DisableLogging
DisableSnapshotCopy
EnableLogging
EnableSnapshotCopy
ModifyCluster
ModifyClusterParameterGroup
ModifyClusterSubnetGroup
ModifyEventSubscription
ModifySnapshotCopyRetentionPeriod
PurchaseReservedNodeOffering

RebootCluster
ResetClusterParameterGroup
RestoreFromClusterSnapshot
RevokeClusterSecurityGroupIngress
RevokeSnapshotAccess
RotateEncryptionKey

## Amazon Route 53 APIs

CreateHostedZone
DeleteHostedZone
ChangeResourceRecordSets
CreateHealthCheck
DeleteHealthCheck
AssociateVPCWithHostedZone
DisassociateVPCFromHostedZone

## Amazon S3 Bucket Level APIs

CreateBucket
DeleteBucket
DeleteBucketCors
DeleteBucketLifecycle
DeleteBucketPolicy
DeleteBucketReplication
DeleteBucketTagging
DeleteBucketWebsite
PutBucketAcl
PutBucketCors
PutBucketLifecycle
PutBucketLogging
PutBucketNotification
PutBucketPolicy
PutBucketReplication

PutBucketRequestPayment
PutBucketTagging
PutBucketVersioning
PutBucketWebsite

## **AWS Storage Gateway APIs**

ActivateGateway
AddCache
AddUploadBuffer
AddWorkingStorage
CancelArchival
CancelRetrieval
CreateCachediSCSIVolume
CreateSnapshot
CreateSnapshotFromVolumeRecoveryPoint
CreateStorediSCSIVolume
CreateTapes
DeleteBandwidthRateLimit
DeleteChapCredentials
DeleteGateway
DeleteSnapshotSchedule
DeleteTape
DeleteTapeArchive
DeleteVolume
DisableGateway
ShutdownGateway
StartGateway
UpdateBandwidthRateLimit
UpdateChapCredentials
UpdateGatewayInformation
UpdateGatewaySoftwareNow
UpdateMaintenanceStartTime

UpdateSnapshotSchedule
------------------------

UpdateVTLDeviceType
---------------------

## Amazon Simple Workflow Service APIs

DeprecateActivityType
-----------------------

DeprecateDomain
-----------------

DeprecateWorkflowType
-----------------------

RegisterActivityType
----------------------

RegisterDomain
----------------

RegisterWorkflowType
----------------------

## Amazon WorkDocs APIs

ActivateUser
--------------

AddUserToGroup
----------------

CreateInstance
----------------

DeactivateUser
----------------

DeleteInstance
----------------

DeregisterDirectory
---------------------

DisableDirectoryMFA
---------------------

EnableDirectoryMFA
--------------------

RegisterDirectory
-------------------

RemoveUserFromGroup
---------------------

ResendUserNotification
------------------------

UpdateDirectoryMFA
--------------------

UpdateInstanceAlias
---------------------

## Resource Types Supported for Event Lookup in CloudTrail

You can look up events by the following resource types for the services indicated.



**AWS CloudTrail User Guide**  
**Resource Types Supported for Event Lookup in**  
**CloudTrail**

**Auto Scaling**

Resource Type	Full syntax
AutoScalingGroup	AWS::AutoScaling::AutoScalingGroup
LaunchConfiguration	AWS::AutoScaling::LaunchConfiguration
ScalingPolicy	AWS::AutoScaling::ScalingPolicy
ScheduledAction	AWS::AutoScaling::ScheduledAction

**AWS CloudTrail**

Resource Type	Full syntax
Trail	AWS::CloudTrail::Trail

**Amazon EC2**

Resource Type	Full syntax
Ami	AWS::EC2::Ami
BundleTask	AWS::EC2::BundleTask
ConversionTask	AWS::EC2::ConversionTask
CustomerGateway	AWS::EC2::CustomerGateway
DHCPOptions	AWS::EC2::DHCPOptions
EIP	AWS::EC2::EIP
EIPAssociation	AWS::EC2::EIPAssociation
ExportTask	AWS::EC2::ExportTask
Instance	AWS::EC2::Instance
InternetGateway	AWS::EC2::InternetGateway
KeyPair	AWS::EC2::KeyPair
NetworkAcl	AWS::EC2::NetworkAcl
NetworkInterface	AWS::EC2::NetworkInterface
NetworkInterfaceAttachment	AWS::EC2::NetworkInterfaceAttachment
PlacementGroup	AWS::EC2::PlacementGroup
ReservedInstance	AWS::EC2::ReservedInstance
ReservedInstancesListing	AWS::EC2::ReservedInstancesListing
ReservedInstancesModification	AWS::EC2::ReservedInstancesModification
RouteTable	AWS::EC2::RouteTable
SecurityGroup	AWS::EC2::SecurityGroup

**AWS CloudTrail User Guide**  
**Resource Types Supported for Event Lookup in**  
**CloudTrail**

Resource Type	Full syntax
Snapshot	AWS::EC2::Snapshot
SpotInstanceRequest	AWS::EC2::SpotInstanceRequest
Subnet	AWS::EC2::Subnet
SubnetNetworkAclAssociation	AWS::EC2::SubnetNetworkAclAssociation
SubnetRouteTableAssociation	AWS::EC2::SubnetRouteTableAssociation
Volume	AWS::EC2::Volume
VPC	AWS::EC2::VPC
VPCPeeringConnection	AWS::EC2::VPCPeeringConnection
VPNConnection	AWS::EC2::VPNConnection
VPNGateway	AWS::EC2::VPNGateway

**Elastic Load Balancing**

Resource Type	Full syntax
LoadBalancer	AWS::ElasticLoadBalancing::LoadBalancer

**IAM**

Resource Type	Full syntax
AccessKey	AWS::IAM::AccessKey
AccountAlias	AWS::IAM::AccountAlias
Group	AWS::IAM::Group
InstanceProfile	AWS::IAM::InstanceProfile
MfaDevice	AWS::IAM::MfaDevice
OpenIDConnectProvider	AWS::IAM::OpenIDConnectProvider
Policy	AWS::IAM::Policy
Role	AWS::IAM::Role
SamlProvider	AWS::IAM::SamlProvider
ServerCertificate	AWS::IAM::ServerCertificate
SigningCertificate	AWS::IAM::SigningCertificate
User	AWS::IAM::User

### Amazon RDS

Resource Type	Full syntax
DBInstance	AWS::RDS::DBInstance
DBOptionGroup	AWS::RDS::DBOptionGroup
DBParameterGroup	AWS::RDS::DBParameterGroup
DBSecurityGroup	AWS::RDS::DBSecurityGroup
DBSnapshot	AWS::RDS::DBSnapshot
DBSubnetGroup	AWS::RDS::DBSubnetGroup
EventSubscription	AWS::RDS::EventSubscription
ReservedDBInstance	AWS::RDS::ReservedDBInstance

### Amazon S3

Resource Type	Full syntax
Bucket	AWS::S3::Bucket

## Controlling User Permissions for CloudTrail

AWS CloudTrail integrates with AWS Identity and Access Management (IAM), which allows you to control access to CloudTrail and to other AWS resources that CloudTrail requires, including Amazon S3 buckets and Amazon Simple Notification Service (Amazon SNS) topics. You can use AWS Identity and Access Management to control which AWS users can create, configure, or delete AWS CloudTrail trails, start and stop logging, and access the buckets that contain log information.

If you work with CloudTrail as the root user in your account, you can perform all the tasks associated with trails, including creating trails, reading logs, and so on. If other people in your organization need to work with CloudTrail, you can create IAM users for those people and give them individual names and passwords. When you do that, you must also give users permissions to work with CloudTrail and with any other AWS services they need to access, such as Amazon S3. (By default, IAM users have no permissions and cannot perform any actions in AWS.)

#### **Important**

We consider it a best practice not to use root account credentials to perform everyday work in AWS. Instead, we recommend that you create an IAM administrators group with appropriate permissions, create IAM users for the people in your organization who need to perform administrative tasks (including for yourself), and add those users to the administrative group. For more information, see [IAM Best Practices](#) in the *IAM User Guide* guide.

#### **Topics**

- [Granting Permissions for CloudTrail Administration](#) (p. 64)
- [Granting Custom Permissions for CloudTrail Users](#) (p. 65)

# Granting Permissions for CloudTrail Administration

To allow users to administer a CloudTrail trail, you must grant explicit permissions to IAM users to perform the actions associated with CloudTrail tasks. For most scenarios, you can do this using an AWS managed policy that contains predefined permissions.

## Note

The permissions you grant to users to perform CloudTrail administration tasks are not the same as the permissions that CloudTrail itself requires in order to deliver log files to Amazon S3 buckets or send notifications to Amazon SNS topics. For more information about those permissions, see [Getting and Viewing Your CloudTrail Log Files \(p. 67\)](#). CloudTrail also requires a role that it can assume to deliver events to an Amazon CloudWatch Logs log group. For more information, see [Granting Custom Permissions for CloudTrail Users \(p. 65\)](#).

A typical approach is to create an IAM group that has the appropriate permissions and then add individual IAM users to that group. For example, you might create an IAM group for users who should have full access to CloudTrail actions, and a separate group for users who should be able to view trail information but not create or change trails.

## To create an IAM group and users for CloudTrail access

1. Open the IAM console at <https://console.aws.amazon.com/iam>.
2. From the dashboard, click **Groups** in the navigation pane, and then click **Create New Group**.
3. Enter a name for the group, and then click **Next Step**.
4. On the **Attach Policy** page, find and select one of the following policies for CloudTrail:
  - **AWSCloudTrailFullAccess**. This policy gives users in the group full access to CloudTrail actions and permissions to manage the Amazon S3 bucket, manage the log group for CloudWatch Logs, and Amazon SNS topic for a trail.
  - **AWSCloudTrailReadOnlyAccess**. This policy lets users in the group view trails and view buckets.

## Note

You can also create a custom policy that grants permissions to individual actions. For details, see [Granting Custom Permissions for CloudTrail Users \(p. 65\)](#).

5. Click **Next Step**.
6. Review the information for the group you are about to create.
  - Note**  
You can edit the group name at this point if you wish, but if you do so, you will have to reselect the policy.
7. Click **Create Group**. The group that you created appears in the list of groups.
8. Click the name of the group that you created, and then click **Add Users to Group**.
9. On the **Add Users to Group** page, select the existing IAM users to add to the new group, and then click **Add Users**. If you don't already have IAM users, click **Create New Users**, enter user names, and then click **Create**.
10. If you created new users, click **Users** in the navigation pane and do the following for each user:
  - a. Select the user.
  - b. If the user will use the console to manage CloudTrail, in the **Security Credentials** tab, click **Manage Password**, and then create a password for the user.

- c. If the user will use the CLI or API to manage CloudTrail, and if you didn't already create access keys, in the **Security Credentials** tab, click **Manage Access Keys** and then create access keys. Store the keys in a secure location.
- d. Give each user his or her credentials (access keys or password).

## Additional Resources

To learn more about creating IAM users, groups, policies, and permissions, see [Creating an Admins Group Using the Console](#) and [Permissions and Policies](#) in the *IAM User Guide* guide.

## Granting Custom Permissions for CloudTrail Users

For most scenarios, the CloudTrail policies grant appropriate permissions to users who work with CloudTrail. However, you might need to grant slightly different permissions to users. If so, you can start by attaching a CloudTrail policy to an IAM group or to an individual user, and then edit the policy to include (or exclude) specific permissions. Or you can create a custom policy and write the policy yourself.

You grant permissions to users, or to groups that users are in, by attaching an [IAM policy](#) to the user or group. (If you used an AWS Managed Policy, you can select the group or user that the policy is attached to, and in the **Permissions** tab, click **Manage Policy**.) Policies are JSON documents that define what actions a user is allowed to perform and what resources the user is allowed to perform those actions on. The following example shows a policy that provides read-only access to CloudTrail trails. It grants permissions to see trail information, but not to create or update trails. The policy also grants permission to read objects in Amazon S3 buckets, but not create or delete them.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudtrail:DescribeTrails",
        "cloudtrail:GetTrailStatus",
        "cloudtrail:LookupEvents",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    }
  ]
}
```

In the policy statements, the `Effect` element specifies whether the actions will be allowed (as here) or denied. The `Action` element lists the specific actions that the user is allowed to perform. The `Resource` element lists the AWS resources that the actions are permitted for. For policies that control access to CloudTrail actions, the `Resource` element is always set to `*`, a wildcard that means "all resources."

Because the policy does not grant permissions for the `CreateTrail`, `UpdateTrail`, `StartLogging` and `StopLogging` actions, the user would not be allowed to turn logging on and off.

The values in the `Action` element correspond to the APIs that the services support. To see what actions CloudTrail supports, see the [AWS CloudTrail API Reference](#). The actions are preceded by `cloudtrail:` to indicate that they refer to CloudTrail actions. For convenience, you can use the `*` wildcard character in actions as well, as shown 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 AWS services.

- `"Action": ["*"]`

This allows all AWS actions. This permission would be suitable for a user who acts as an AWS administrator for your account.

The following example shows a "full access" (administrator) policy for working with CloudTrail. This policy lets a user perform all CloudTrail actions. It also lets the user manage files in Amazon S3 buckets, manage CloudWatch Logs monitoring of CloudTrail log events, and manage Amazon SNS topics in the account that the user is associated with.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:AddPermission",
        "sns:CreateTopic",
        "sns>DeleteTopic",
        "sns:ListTopics",
        "sns:SetTopicAttributes"
      ],
      "Resource": "arn:aws:sns:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3>DeleteBucket",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy"
      ],
      "Resource": "arn:aws:s3::*"
    },
    {
      "Effect": "Allow",
      "Action": "cloudtrail:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogGroup",
    ],
    "Resource": "arn:aws:logs:*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole",
      "iam:ListRoles",
      "iam:GetRolePolicy",
    ],
    "Resource": "arn:aws:iam::*"
  }
]
```

## Additional Resources

To learn more about creating IAM users, groups, policies, and permissions, see [Creating an Admins Group Using the Console](#) and [Permissions and Policies](#) in the *IAM User Guide* guide.

# Getting and Viewing Your CloudTrail Log Files

After you've set up CloudTrail to capture the log files you want, you'll 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. Typically, log files appear in your bucket within 15 minutes of the recorded AWS API call or other AWS event. Log files are generally published every five minutes.

### Topics

- [Finding Your CloudTrail Log Files](#) (p. 67)
- [Reading Your CloudTrail Log Files](#) (p. 68)

## Finding Your CloudTrail Log Files

CloudTrail publishes log files to your S3 bucket in a gzip archive. Within the S3 bucket, the log file has a formatted name that includes the following elements:

- The bucket name that you specified when creating your trail
- The (optional) prefix you specified when creating 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/re  
gion/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. For example, open the Amazon S3 console, click on the name of the bucket in which you're interested, and keep clicking through the object hierarchy until you get to the log file you're looking for. All log files have a .gz extension. You will be navigating through an object hierarchy that is similar to the following 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 it is quite rare, 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 \(p. 144\)](#).

## Reading Your CloudTrail Log Files

This topic describes options for retrieving and viewing your CloudTrail log files.

### Retrieving your log files

CloudTrail log files are Amazon S3 objects. You can retrieve them by using the Amazon S3 console, the AWS command line interface (CLI), or the Amazon S3 API. For more information, see [Working with Amazon S3 Objects](#) in the Amazon Simple Storage Service Developer Guide. The Amazon Simple Storage Service Console User Guide covers using the console to retrieve your objects. For example, open the Amazon S3 console, click on the name of the bucket in which you're interested, and keep clicking through the object hierarchy until you get to the log file you're looking for. All log files have a .gz extension.

### Viewing your log files

Log files are written in JSON (JavaScript Object Notation) format. If you have a JSON viewer add-on installed, you can view the files directly in your browser by double-clicking the log file name in the Amazon S3 bucket. This will open a new window or a new tab, depending on the add-on and on the browser, that displays the JSON in a readable format. To find a JSON viewer, search on that phrase in your browser of choice.



For example, if you use Google Chrome in Windows or Linux, a popular extension that you can add is named [JSONView](#) which you can download from the Chrome Web Store. 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 Amazon S3 bucket to open the log file in JSON format. There is no comparable extension for Internet Explorer, but there is a [registry edit](#) you can make to enable Internet Explorer to open JSON files after you download and decompress them.

An alternate approach to viewing your CloudTrail logs on Windows is to download them locally and use a text editor such as [Notepad++](#) along with the [JSON Viewer](#) plug-in. To download a log file, right-click on the file in your Amazon S3 bucket and right-click **Download** in the pop-up window. Click **Save link as...** and follow the prompts to save the file locally. This saves the file, however, in compressed format. You must use a product such as [7-Zip](#) to extract the uncompressed JSON data. After decompressing the file, open it in Notepad++, select all of the text, and navigate to **Plugins**, point to **JSON Viewer**, and then click **Format JSON**.

For more information about the event fields that can appear in a log file entry, see [CloudTrail Event Reference](#) (p. 143).

AWS partners with third-party specialists in logging and analysis to provide solutions that leverage CloudTrail output. For more information, visit the CloudTrail detail page at <http://aws.amazon.com/cloudtrail>.

**Note**

For log files captured during the last seven days, you can use the CloudTrail console, the AWS CLI or the AWS SDKs. For more information, see [Viewing CloudTrail Events](#) (p. 30).

## 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. 69)
- [CloudTrail Permissions for SNS Notifications](#) (p. 70)

## Configuring CloudTrail to Send Notifications

When you set up a trail, CloudTrail can configure a new Amazon SNS topic for you. If you create your trail using either the CloudTrail console or the `aws cloudtrail create-subscription` CLI command, and if you specify that you want notifications, CloudTrail creates the Amazon SNS topic for you and attaches an appropriate policy to it to allow CloudTrail to publish to that topic.

A CloudTrail trail works in a specific region. When you enable notifications, notifications are sent to an Amazon SNS topic in that region. If you have CloudTrail enabled for multiple regions, you need to create a separate topic for each region and subscribe to them individually.

CloudTrail lets you configure a trail to send notifications to an Amazon SNS topic, but it does not manage subscriptions for you. In order to receive notifications, you must subscribe to the Amazon SNS topic or

topics that CloudTrail uses. You do this using the Amazon SNS console or Amazon SNS commands. For information, see [Subscribe to a Topic](#) in the *Amazon Simple Notification Service Developer Guide*.

**Tip**

Because CloudTrail sends a notification each time a log file is written to the Amazon S3 bucket, an account that's very active can generate a large number of notifications. If you subscribe using email or SMS, you can end up receiving 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 [Subscribe Queue to 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:

```
"Message" :    "{\"s3Bucket\": \"<your_bucket_name>\", \"s3ObjectKey\" :  
[\"AWSLogs/279630728954/CloudTrail/us-west-2/2013/12/13/<your_account_num  
ber>_CloudTrail_us-west-2_20131213T1920Z_LnPgDQnpkSKEspV.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 discussion of the JSON structure, see the topic [Sending Amazon SNS Messages to Amazon SQS Queues](#) in the *Amazon Simple Notification Service Developer Guide*. Only the **Message** field conveys CloudTrail information. The other fields contain information from the Amazon SNS service itself.

If you create a trail using the API, you can specify an existing Amazon SNS topic that you want CloudTrail to send notifications to. In that case, you must make sure that the topic exists and that it has permissions that let CloudTrail send notifications to it. (See below.) You can specify that you want CloudTrail to use an existing Amazon SNS topic by calling the [CreateTrail](#) or [UpdateTrail](#) operations, which are part of the CloudTrail API.

## Additional Resources

To learn more about creating Amazon SNS topics and about subscribing to them, see the [Amazon Simple Notification Service Developer Guide](#).

## CloudTrail Permissions for SNS Notifications

CloudTrail must have permissions to send notifications to an Amazon SNS topic. If CloudTrail creates the topic for you automatically (for example, if you use the console to set up a new trail or use the `aws cloudtrail create-subscription` command), these permissions are automatically added to the new topic. However, if you specify an existing topic, you must make sure that the topic has the correct permissions.

The following example shows the permissions that are automatically created by CloudTrail for a new topic. This policy statement allows CloudTrail to publish to a specified Amazon SNS topic.

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Sid": "AWSCloudTrailSNSPolicy20140219",  
    "Effect": "Allow",  
    "Principal": {"AWS": [  
      "arn:aws:iam::903692715234: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",
        "arn:aws:iam::035351147821:root"
    ]},
    "Action": "SNS:Publish",
    "Resource": "arn:aws:sns:region:account-number:topic-name"
}
}]
}
```

In the `Resource` field, *account-number* is the account number of the topic owner; in topics that you create, this will be your account number. You must substitute appropriate values for *region* and *topic-name*.

In the `Principal` field, the account number must match the CloudTrail account for that region. For more information about the supported regions and their associated account numbers, see [CloudTrail Supported Regions](#) (p. 13).

## Additional Resources

To learn more about creating Amazon SNS topics and about subscribing to them, see the [Amazon Simple Notification Service Developer Guide](#).

# Working with CloudTrail Log Files

---

The following topics discuss how to perform more complex actions with your CloudTrail files. You can aggregate your log files into a single S3 bucket from multiple regions or accounts. You can share customer log files between accounts. You can monitor CloudTrail log files in real time by sending them to CloudWatch Logs. You can also use the AWS CloudTrail Processing Library to write log processing applications in Java.

## Topics

- [Receiving CloudTrail Log Files from Multiple Sources in a Single Amazon S3 Bucket](#) (p. 72)
- [Sharing CloudTrail Log Files Between AWS Accounts](#) (p. 81)
- [Monitoring CloudTrail Log Files with Amazon CloudWatch Logs](#) (p. 92)
- [Using the CloudTrail Processing Library](#) (p. 135)

## Receiving CloudTrail Log Files from Multiple Sources in a Single Amazon S3 Bucket

With AWS CloudTrail, you can choose to have log files from multiple AWS regions and multiple AWS accounts delivered to a single Amazon S3 bucket. Receiving your log files in this way simplifies storing and managing them. For more information, see the following topics.

## Topics

- [Receiving CloudTrail Log Files from Multiple Regions](#) (p. 72)
- [Receiving CloudTrail Log Files from Multiple Accounts](#) (p. 78)

## Receiving CloudTrail Log Files from Multiple Regions

To receive CloudTrail log files from multiple regions in a single Amazon S3 bucket, you must complete the following steps in order.

1. Turn on CloudTrail in your AWS account, making sure to do so in only one region. When you turn on CloudTrail in your AWS account, it's a good idea to accept the default and choose to create a new Amazon S3 bucket for your log files. Then, when you turn on CloudTrail in additional regions, you can

use the same log file prefix that you used when you turned on CloudTrail in the original region. This way, your log files from all regions are placed into a single bucket without your having to manually update the bucket permissions. For more information, see [Creating a Trail \(p. 26\)](#).

2. Ensure that the bucket policy on your destination bucket grants the necessary permissions to CloudTrail. If you chose to create a new Amazon S3 bucket during the preceding step, the necessary permissions are granted for you. If, however, you specified an existing bucket during the previous step, you must grant the necessary permissions by applying a bucket policy. For more information, see [Setting Bucket Policy for CloudTrail Manually \(p. 73\)](#).
3. Turn on CloudTrail in the other regions where you have AWS resources, and then configure CloudTrail to use the same bucket and the same log file prefix that you specified in step 1. For more information, see [Turning On CloudTrail in Additional Regions \(p. 74\)](#).

### Topics

- [Setting Bucket Policy for CloudTrail Manually \(p. 73\)](#)
- [Turning On CloudTrail in Additional Regions \(p. 74\)](#)
- [Turning on CloudTrail in All Regions with the AWS CLI \(p. 75\)](#)

## Setting Bucket Policy for CloudTrail Manually

This procedure is necessary **only** if you chose to use an existing Amazon S3 bucket when you first turned on the CloudTrail service. If you chose to create a new bucket when you turned on CloudTrail, then the CloudTrail service has already applied the permissions necessary to deliver log files to the destination bucket.

### To attach the policy required by CloudTrail to an Amazon S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket where you want CloudTrail to deliver your log files and then click **Properties**.
3. Click **Permissions**.
4. Click **Edit Bucket Policy**.
5. Copy the following policy into the **Bucket Policy Editor** window and then substitute the correct names of your bucket, prefix, and account ID for the placeholders indicated in italics. Your AWS account ID is a twelve-digit number, and leading zeros must not be omitted.

#### Caution

If the existing bucket already has one or more policies attached to it, add the statements for CloudTrail access to that policy or policies. Take care to evaluate the resulting set of permissions to be sure that they are appropriate for the users who will be accessing the bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck20131101",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam:903692715234: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",
        "arn:aws:iam::035351147821:root"
    ]
  },
  "Action": "s3:GetBucketAcl",
  "Resource": "arn:aws:s3:::myBucketName"
},
{
  "Sid": "AWSCloudTrailWrite20131101",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::903692715234: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",
      "arn:aws:iam::035351147821:root"
    ]
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::myBucketName/[optional] myLogFilePrefix/AWS
Logs/myAccountID/*",
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}
]
}
```

**Note**

In the `Principal` fields, the account number must match the CloudTrail account for that region. For more information about the supported regions and their associated account numbers, see [CloudTrail Supported Regions \(p. 13\)](#).

## Turning On CloudTrail in Additional Regions

Turning on CloudTrail in an additional region is similar to the steps you took to turn it on in the first region in which you created a trail, with the following differences:

- Instead of creating a new S3 bucket, you'll most likely want to use your existing S3 bucket to receive the log files from the new region.
- Global services generate the same events in all regions, so to avoid receiving duplicate global service events in your log files, you will configure the new region to not include global service events.
- If you are writing to an existing bucket and use a log file prefix that is different from the one in the first region, you must edit the policy on the bucket to allow CloudTrail to write logs that have the new prefix to that bucket.

- Amazon SNS is a regional service, so if you enable SNS notifications in the new region, you will need to subscribe to the CloudTrail SNS topic in the new region.

### To turn on CloudTrail in additional regions

1. Sign into the AWS management console. Open the AWS CloudTrail console. In the navigation bar, select the region where you want to turn on CloudTrail. This should be a different region than that which you chose when you initially created a trail.
2. Click **Get Started Now**.
3. On the following page, for **Create a new S3 bucket?**, select **No**. Select an S3 bucket that exists in the current account or use the text box to enter the name of an existing bucket that is not in your current account. Remember that you must manually edit the bucket policy to grant CloudTrail permission to write to it. For more information, see [Setting Bucket Policy for CloudTrail Manually \(p. 73\)](#).
4. Click **Advanced**.
5. In the **Log file prefix** field, enter the same prefix you entered when you turned on CloudTrail in the first region.

#### Note

If you choose to use a prefix that is different from the one you entered when you turned on CloudTrail in the first region, you must edit the bucket policy on your destination bucket to allow CloudTrail to write log files to your bucket using this new prefix.

6. (Optional) Select **Yes** or **No** for **SNS notification for every log file delivery?** If you select **Yes**, enter a name for your Amazon SNS topic in the **SNS topic (new)** field.

#### Note

Amazon SNS is a regional service, so if you choose to create a topic, that topic will exist in the same region in which you are enabling CloudTrail. If you have enabled CloudTrail's SNS notification in more than one region, you need to subscribe to CloudTrail's SNS topic in each region where one exists. See the [Amazon Simple Notification Service Getting Started Guide](#) for more information.

7. Click **Turn On**.
8. Navigate to the **Configuration** page of the CloudTrail console.
9. Click the pencil icon to the right of **Additional Configuration** to edit the **Include global services** option.
10. For this additional region, select **No** for **Include global services?** to avoid recording duplicate API calls from global services such as IAM or AWS STS. (Normally you select **Yes** for the first region where CloudTrail is enabled, and **No** for subsequent regions.)

#### Note

Global services generate the same events in all regions, so selecting **Yes** for regions other than your first one would result in duplicate entries for the global events in the log files.

In about 15 minutes, CloudTrail starts publishing log files that show the AWS calls made in your accounts in this region since you completed the preceding steps.

## Turning on CloudTrail in All Regions with the AWS CLI

Instead of turning on CloudTrail (that is, "creating trails") in all regions one by one, you can use the AWS CLI from either the Linux command line or Windows PowerShell.

## Prerequisites

- The AWS CLI command line tools. For more information, see the [AWS Command Line Interface User Guide](#).
- A preexisting S3 bucket that will receive your CloudTrail logs from all regions.
- The bucket that you specify must already have a policy attached to it that allows CloudTrail to write to it. For more information, see [Setting Bucket Policy for CloudTrail Manually](#) (p. 73).
- Decide upon the region that will receive global service events. To avoid duplication of global service event logs in your bucket, the commands include global service events only in the region that you specify.

## Using Linux to Create Trails in All Regions

These commands first create variables to store your AWS CLI profile name, the name of your S3 bucket, and the region that will receive global service events (this example uses `us-east-1`). Next, Amazon EC2 and the AWS CLI are used to create a list of regions. The last command iterates through that list, turning on CloudTrail and starting logging in each region. Global service events are included only in the region specified, and each trail is named after its corresponding region. For example, the trail in US West (Oregon) is named `us-west-2`. For information on named profiles in the AWS CLI, see [Named Profiles](#).

### Note

Depending on your environment, you may have to prefix `aws` with the path to the binaries for the AWS CLI (for example, `./bin/aws`).

```
# Create trails in all AWS standard regions with the AWS CLI and Linux.
PROFILE="default"
CLOUDTRAIL_S3_BUCKET="your_S3_bucket_name"
REGION_FOR_GLOBAL_EVENTS="us-east-1"
regionlist=$(aws ec2 describe-regions --query Regions[*].RegionName --output
text)
for region in "${regionlist[@]}"
do
if
[ $region = $REGION_FOR_GLOBAL_EVENTS ]
then
aws --profile $PROFILE --region $region cloudtrail create-trail --name $region
--s3-bucket-name $CLOUDTRAIL_S3_BUCKET --include-global-service-events --output
table
else
aws --profile $PROFILE --region $region cloudtrail create-trail --name $region
--s3-bucket-name $CLOUDTRAIL_S3_BUCKET --no-include-global-service-events --
output table
fi
aws --profile $PROFILE --region $region cloudtrail start-logging --name $region
--output table
done
```

## Using Linux to Delete Trails in All Regions

The same approach can be used to delete trails in all regions. The sample commands rely on the convention that the name of the trail in each region is the same as the region name (for example, that the trail in US West (Oregon) is named `us-west-2`). Unlike `create-trail`, `delete-trail` provides no feedback, so the example command echoes the deletion of each region's trail to the command window.



```
$ regionlist=$(aws ec2 describe-regions --query Regions[*].RegionName --output text)

$ for region in "${regionlist[@]}"; do aws --region $region cloudtrail delete-trail --name $region; echo "Trail in" $region "deleted."; done
```

## Using Windows PowerShell to Create Trails in All Regions

These commands first create variables to store your AWS CLI profile name, the name of your S3 bucket, and the region that will receive global service events (this example uses `us-east-1`). Next, Amazon EC2 and the AWS CLI are used to create a list of regions. The last command iterates through that list, turning on CloudTrail and starting logging in each region. Global service events are included only in the region specified, and each trail is named after its corresponding region. For example, the trail in US West (Oregon) is named `us-west-2`. For information on named profiles in the AWS CLI, see [Named Profiles](#).

```
# Create trails in all AWS standard regions with the AWS CLI and PowerShell.

$profile = "default"

$cloudtrails3bucket = "your_S3_bucket_name"

$regionforglobalevents = "us-east-1"

$regionlist = aws ec2 describe-regions --query Regions[*].RegionName --output text

$regionlist = $regionlist -split '\s+' | sort

Foreach ($region in $regionlist)
{
    If ($region -eq $regionforglobalevents)
    {
        aws --profile $profile --region $region cloudtrail create-trail --
name $region --s3-bucket-name $cloudtrails3bucket --include-global-service-
events --output table
    }

    Else
    {
        aws --profile $profile --region $region cloudtrail create-trail --
name $region --s3-bucket-name $cloudtrails3bucket --no-include-global-service-
events --output table
    }

    aws --profile $profile --region $region cloudtrail start-logging --name
$region --output table
}
```

## Using Windows PowerShell to Delete Trails in All Regions

The same approach can be used to delete trails in all regions. The sample commands rely on the convention that the name of the trail in each region is the same as the region name (for example, that

the trail in US West (Oregon) is named `us-west-2`). Unlike `create-trail`, `delete-trail` provides no feedback, so the example command echoes the deletion of each region's trail to the command window.

```
$regionlist = aws ec2 describe-regions --query Regions[*].RegionName --output text
$regionlist = $regionlist -split '\s+' | sort

Foreach ($region in $regionlist)
{
Write-Host -NoNewLine "Deleting trail in region" $region
Write-Host
aws --region $region cloudtrail delete-trail --name $region
}
```

## Receiving CloudTrail Log Files from Multiple Accounts

You can have CloudTrail deliver log files from multiple AWS accounts into a single Amazon S3 bucket. For example, you have four AWS 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 \(p. 26\)](#).

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 \(p. 78\)](#).

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 \(p. 80\)](#).

### Topics

- [Setting Bucket Policy for Multiple Accounts \(p. 78\)](#)
- [Turning on CloudTrail in Additional Accounts \(p. 80\)](#)

## 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.

### To modify bucket permissions so that files can be received from multiple accounts

1. Sign in to the AWS Management Console using the account that owns the bucket (111111111111 in this example) and open the Amazon S3 console.
2. Select the bucket where CloudTrail delivers your log files and then click **Properties**.

3. Click **Permissions**.
4. Click **Edit Bucket Policy**.
5. 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.

**Note**

An AWS account ID is a twelve-digit number, and leading zeros must not be omitted.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCloudTrailAclCheck20131101",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::903692715234: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",
          "arn:aws:iam::035351147821:root"
        ]
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::myBucketName"
    },
    {
      "Sid": "AWSCloudTrailWrite20131101",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::903692715234: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",
          "arn:aws:iam::035351147821:root"
        ]
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::myBucketName/[optional] myLogFilePrefix/AWS
Logs/111111111111/*",
        "arn:aws:s3:::myBucketName/[optional] myLogFilePrefix/AWS
Logs/222222222222/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

```
}  
  }  
} ]  
}
```

**Note**

In the `Principal` fields, the account number must match the CloudTrail account for that region. For more information about the supported regions and their associated account numbers, see [CloudTrail Supported Regions \(p. 13\)](#).

## Turning on CloudTrail in Additional Accounts

You can use the console or the command line interface to turn on CloudTrail in additional AWS accounts. For more information, see the following topics:

**Topics**

- [Using the Console to Turn on CloudTrail in Additional AWS Accounts \(p. 80\)](#)
- [Using the CLI to Turn on CloudTrail in Additional AWS Accounts \(p. 81\)](#)

## Using the Console to Turn on CloudTrail in Additional AWS Accounts

You can use the CloudTrail console to turn on CloudTrail in additional accounts.

1. Sign into the AWS management console using account 222222222222 credentials and open the AWS CloudTrail console. In the navigation bar, select the region where you want to turn on CloudTrail.
2. Click **Get Started Now**.
3. On the following page, for **Create a new S3 bucket?**, select **No**. 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. CloudTrail displays a warning asking you if you are sure that you want to specify an S3 bucket in another account. Verify the name of the bucket you entered, and if you are sure it's the correct one, click **Continue**.
4. Click **Advanced**.
5. In the **Log file prefix** field, 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.
6. (Optional) Select **Yes** or **No** for **SNS notification for every log file delivery?**. If you select **Yes**, enter a name for your Amazon SNS topic in the **SNS topic (new)** field.

**Note**

Amazon SNS is a regional service, so if you choose to create a topic, that topic will exist in the same region in which you are enabling CloudTrail. If you have turned on CloudTrail SNS notification in more than one region, you need to subscribe to the CloudTrail SNS topic in each region. See the Amazon Simple Notification Service Getting Started Guide for more information.

7. Click **Turn On**.

In about 15 minutes, CloudTrail starts publishing log files that show the AWS calls made in your accounts in this region since you completed the preceding steps.

## Using the CLI to Turn on CloudTrail in Additional AWS Accounts

You can use the AWS 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 [AWS Command Line Interface User Guide](#).

Turn on CloudTrail in your additional accounts by using the `create-subscription` command. Use the following options to specify additional settings:

- `--name` specifies the name of the trail.
- `--s3-use-bucket` specifies the existing Amazon S3 bucket, 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).
- `--sns-new-topic` specifies the name of the Amazon SNS topic to which you can subscribe for notification of log file delivery to your bucket (optional).

In contrast to trails that you create using the console, you must give every trail you create with the AWS CLI a name. You can create one trail for each region in which an account is running AWS resources.

The following example command shows how to create a trail for your additional accounts by using the AWS 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-new-bucket` option. Amazon S3 bucket names are globally unique.

```
aws cloudtrail create-subscription --name AWSCloudTrailExample --s3-use-bucket  
MyBucketBelongingToAccount111111111111 --s3-prefix AWSCloudTrailPrefixExample  
--sns-new-topic AWSCloudTrailLogDeliveryTopicExample
```

When you run the command, you will see output similar to the following:

```
CloudTrail configuration:  
{  
  "trailList": [  
    {  
      "S3KeyPrefix": "AWSCloudTrailPrefixExample",  
      "IncludeGlobalServiceEvents": true,  
      "Name": "AWSCloudTrailExample",  
      "SnsTopicName": "AWSCloudTrailLogDeliveryTopicExample",  
      "S3BucketName": "MyBucketBelongingToAccount111111111111"  
    }  
  ]  
}
```

For more information about using CloudTrail from the AWS command line tools, see the [CloudTrail command line reference](#).

## Sharing CloudTrail Log Files Between AWS Accounts

This section explains how to share CloudTrail log files between multiple AWS accounts. We will assume that the log files have all been received in a single Amazon S3 bucket. You do not need to receive log files in this way in order to share them, but [receiving CloudTrail log files in a single Amazon S3 bucket](#)

**AWS CloudTrail User Guide**  
**Scenario 1: Granting Access to the Account that**  
**Generated the Log Files**

---

across accounts and regions is quite common and we will assume that you have done so when discussing the two scenarios that follow. 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 AWS 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 an IAM 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 [Roles \(Delegation and Federation\)](#) in *IAM User Guide*.

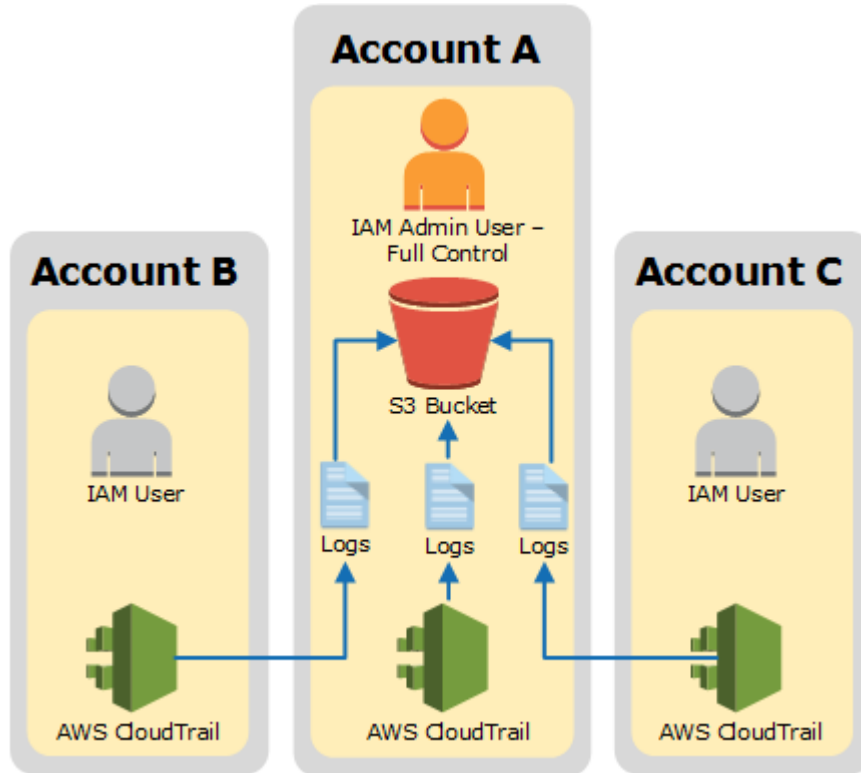
## Scenario 1: Granting Access to the Account that Generated the Log Files

In this scenario, we'll assume that your enterprise is made up of two business units and that it maintains three AWS 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.

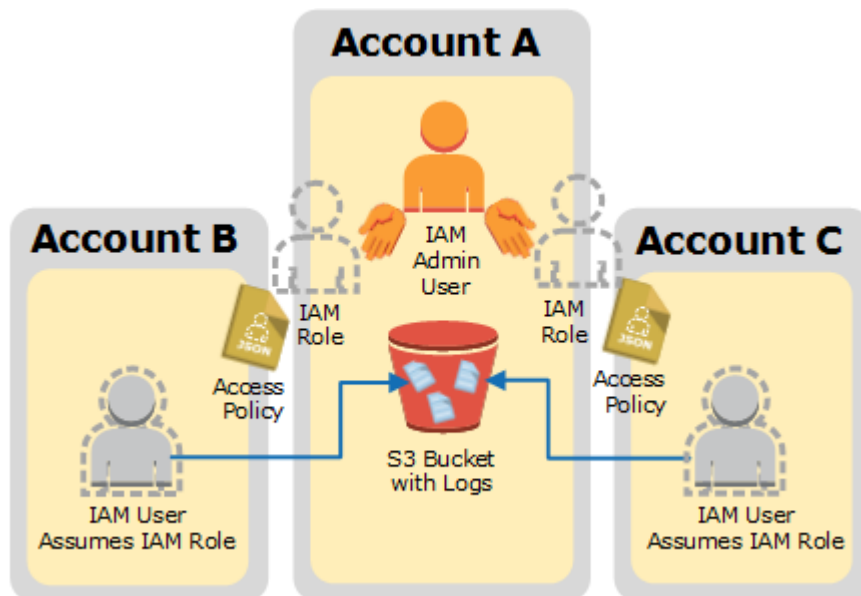
This scenario assumes that you have already configured the log files from all three accounts to be delivered to a single Amazon S3 bucket, and that account A has full control over that bucket, as shown in the following illustration.

**AWS CloudTrail User Guide**  
**Scenario 1: Granting Access to the Account that**  
**Generated the Log Files**

---



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. How: [Creating a Role \(p. 85\)](#)

- 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. How: [Creating an Access Policy to Grant Access to Accounts You Own \(p. 86\)](#)
- 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. How: [Creating permissions policies for IAM users \(p. 90\)](#).
- 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. How: [Calling AssumeRole \(p. 90\)](#).

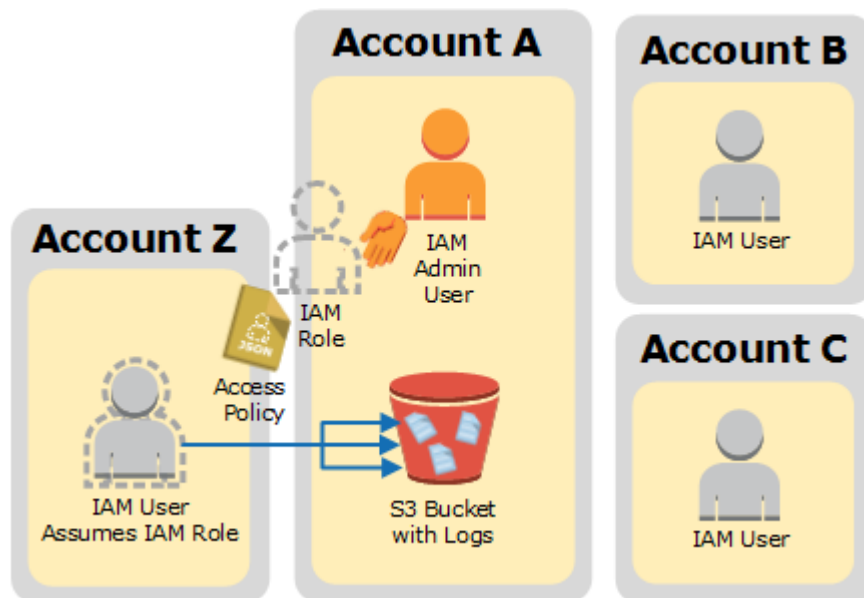
The IAM users in accounts B and C can then programmatically retrieve their own log files, but not the log files of any other account.

## 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 AWS 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 AWS 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 \(p. 85\)](#)



- 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. 88)
- Have an IAM user in Account Z programmatically assume the role and then retrieve the appropriate log files. The IAM user must be given permission to assume the role by the owner of Account Z. How: [Creating permissions policies for IAM users](#) (p. 90). 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. 90).

### Topics

- [Creating a Role](#) (p. 85)
- [Creating an Access Policy to Grant Access to Accounts You Own](#) (p. 86)
- [Creating an Access Policy to Grant Access to a Third Party](#) (p. 88)
- [Assuming a Role](#) (p. 89)
- [Stop Sharing CloudTrail Log Files Between AWS Accounts](#) (p. 91)

## 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 AWS 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 [Roles \(Delegation and Federation\)](#). For more information about how to assume a role, see [Assuming a Role](#) (p. 89).

## 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 an IAM user with the proper permissions in those accounts can assume the role.

The *trust policy* is automatically created 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 CLI to create the role, you must specify the trust policy in the `create-role` 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 that 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. 86). For Scenario 2 that 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. 88).

For further details about creating and working with IAM policies, see [Permissions and Policies](#) in *IAM User Guide*.

## Creating a Role

### To Create a Role by Using the Console

1. Sign into the AWS Management Console as an administrator of Account A.
2. Navigate to the IAM console.
3. In the navigation pane, click **Roles**.
4. Click **Create New Role**.
5. Enter a name for the new role, and then click **Next Step**.
6. Click **Role for Cross-Account Access**.
7. For Scenario 1, do the following to provide access between accounts you own:
  - a. Select **Provide access between AWS accounts you own**.
  - b. Enter the twelve-digit account ID of the account (B, C, or Z) to be granted access.
  - c. 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. Select **Allows IAM users from a 3rd party AWS account to access this account**.
  - b. Enter the twelve-digit account ID of the account (Account Z) to be granted access.
  - c. Enter an external ID that provides additional control over who can assume the role. For more information, see [About the External ID](#) in the [AWS Security Token Service User Guide](#).
8. Click **Next Step** to attach a policy that sets the permissions for this role.
  9. Under **Attach Policy**, find and select 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. 86).
  - 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. 88).
10. Click **Next Step**
  11. Review the role information.

#### Note

You can edit the role name at this point if you wish, but if you do so, you will be taken back to the **Step 2: Select Role Type** page where you must reenter the information for the role.

12. Click **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

## AWS CloudTrail User Guide

### Creating an Access Policy to Grant Access to Accounts You Own

---

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 \(p. 80\)](#).

#### 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 have created roles for each account and specified 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 have created roles for each account and specified 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 \(p. 89\)](#).

## 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. The trust relationship, automatically created by AWS when you create the role, 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 \(p. 85\)](#).

For example, the trust relationship created by AWS will specify that Account Z is trusted to assume the role created by Account A, and will look something like the following.

```
{
  "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 **Condition** element is shown in the following example access policy. For more information, see [About the External ID](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-z-id:root"
      },
      "Condition": {
        "StringEquals": {
          "sts:AssumeRoleExternalID": "external-id"
        }
      }
    }
  ]
}
```

```
    "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. The wild card (\*) at the end of the **Resource** value indicates that Account Z can access any log file in the Amazon 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 have created a role for Account Z and specified 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 Amazon S3 bucket. For more information, see [Assuming a Role \(p. 89\)](#).

## Assuming a Role

You must designate a separate IAM user to assume each role you've created in each account, and ensure that each IAM user has appropriate permissions.

## IAM Users and Roles

After you have created 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 will programmatically assume the appropriate role to access the log files. That is, the user in account B will assume the role created for account B, the user in account C will assume the role created for account C, and the user in account Z will assume the role created for account Z. When a user assumes a role, AWS returns temporary security

credentials that can be used to make requests to list, retrieve, copy, or delete the log files depending on the permissions granted by the access policy associated with the role.

For more information about working with IAM users, see [Working with IAM Users and Groups](#) .

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 \(p. 86\)](#).
- 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 \(p. 88\)](#).

## Creating permissions policies for IAM users

To perform the actions permitted by the roles, the IAM user must have permission to call the AWS STS [AssumeRole](#) API. You must edit the *user-based policy* for each IAM user to grant them the appropriate permissions. That is, you set a **Resource** element in the policy that is attached 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.

### To attach the required policy to the IAM role

1. Sign in to the AWS Management Console and open the IAM console.
2. Select the user whose permissions you want to modify.
3. Select the **Permissions** tab.
4. Select **Custom Policy**.
5. Select **Use the policy editor to customize your own set of permissions**.
6. Give the policy a name.
7. Copy the following policy into the space provided for the policy document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::account-A-id:role/Test"
    }
  ]
}
```

### Important

Only IAM users can assume a role. If you attempt to use AWS root account credentials to assume a role, access will be denied.

## Calling AssumeRole

A user in accounts B, C, or Z can assume a role by creating an application that calls the AWS 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

## AWS CloudTrail User Guide

### Stop Sharing CloudTrail Log Files Between AWS Accounts

---

during role creation. For more information, see [About the External ID](#) . You can retrieve the ARN from the Account A by opening the IAM console.

#### To Find the ARN Value in Account A

- Select **Roles**
- Select the role you want to examine.
- Look for the Role ARN in the **Summary** tab

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 [AWS 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.

```
import boto
from boto.sts import STSConnection
from boto.s3.connection import S3Connection

# The calls to AWS STS AssumeRole must be signed using the access key ID and
secret
# access key of an IAM user or using existing temporary credentials. (You cannot
call
# AssumeRole using the access key for an account.) The credentials can be in
# environment variables or in a configuration file and will be discovered
automatically
# by the STSConnection() function. For more information, see the Python SDK
# documentation: http://boto.readthedocs.org/en/latest/boto_config_tut.html

sts_connection = STSConnection()
assumedRoleObject = sts_connection.assume_role(
    role_arn="arn:aws:iam::account-of-role-to-assume:role/name-of-role",
    role_session_name="AssumeRoleSession1"
)

# Use the temporary credentials returned by AssumeRole to call Amazon S3
# and list the bucket in the account that owns the role (the trusting account)
s3_connection = S3Connection(
    aws_access_key_id=assumedRoleObject.credentials.access_key,
    aws_secret_access_key=assumedRoleObject.credentials.secret_key,
    security_token=assumedRoleObject.credentials.session_token
)
bucket = s3_connection.get_bucket(bucketname)
print bucket.name
```

## Stop Sharing CloudTrail Log Files Between AWS Accounts

To stop sharing log files to another AWS account, simply delete the role that you created for that account in [Creating a Role](#) (p. 85).

1. Sign in to the AWS Management Console as an IAM user with administrative-level permissions for Account A.
2. Navigate to the IAM console.
3. In the navigation pane, click **Roles**.
4. Select the role you want to delete.
5. Right-click and select **Delete Role** from the context menu.

## Monitoring CloudTrail Log Files with Amazon CloudWatch Logs

### Topics

- [Sending CloudTrail Events to CloudWatch Logs](#) (p. 92)
- [Using an AWS CloudFormation Template to Create CloudWatch Alarms](#) (p. 95)
- [Creating CloudWatch Alarms for CloudTrail Events: Examples](#) (p. 104)
- [Creating CloudWatch Alarms for CloudTrail Events: Additional Examples](#) (p. 127)
- [Configuring Notifications for CloudWatch Logs Alarms](#) (p. 133)
- [Stopping CloudTrail from Sending Events to CloudWatch Logs](#) (p. 133)
- [CloudWatch Log Group and Log Stream Naming for CloudTrail](#) (p. 134)
- [Role Policy Document for CloudTrail to Use CloudWatch Logs for Monitoring](#) (p. 134)

One of the ways that you can work with CloudTrail logs is to monitor them in real time by sending them to CloudWatch Logs. You can configure CloudTrail to send log files to a CloudWatch Logs log group. You define CloudWatch Logs metric filters that will evaluate your CloudTrail log events for matches in terms, phrases, or values. You assign CloudWatch metrics to the metric filters. You also create CloudWatch alarms that are triggered according to thresholds and time periods that you specify. You can configure an alarm to send a notification when the alarm is triggered so that you can take immediate action. You can also configure CloudWatch to automatically perform an action in response to an alarm. CloudTrail events are protected by SSL encryption as they are delivered from CloudTrail to the CloudWatch Logs log group.

CloudWatch Logs is supported in the `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `eu-central-1`, `eu-west-1`, `us-east-1`, `us-west-1`, and `us-west-2` regions. Standard pricing for Amazon CloudWatch and Amazon CloudWatch Logs apply. For more information, see [Amazon CloudWatch Pricing](#).

## Sending CloudTrail Events to CloudWatch Logs

This topic describes how to configure CloudTrail to send logs to CloudWatch Logs so that you can monitor CloudTrail log events. Sending CloudTrail events to a CloudWatch Logs log group requires you to do a minimum of the following:

- Create a log group or specify an existing one.
- Specify or create an IAM role.
- Attach a role policy, or use the default.

### Note

You must create a trail before you can configure CloudTrail to send log events to CloudWatch Logs. The procedures in this section assume you have already created a trail. For more information, see [Creating a Trail](#) (p. 26) or [Creating and Updating a Trail with the AWS CLI](#)



(p. 27). Also make sure to configure permissions for your IAM user to create a log group, change the log group, and assume the role by creating an action policy. For more information, see [Granting Custom Permissions for CloudTrail Users](#) (p. 65).

This topic includes procedures for the AWS Management Console and the AWS Command Line Interface (AWS CLI).

## Configuring CloudWatch Logs Monitoring Using the Console

You can use the AWS Management Console to configure CloudTrail to send log 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 new log group or specify an existing one.

#### To specify a log group using the console

1. Navigate to the **CloudTrail Configuration** page.
2. In the **CloudWatch Logs (Optional)** box, do one of the following:
  - a. If you do not yet have any CloudWatch logs configured, click **Configure**.
  - b. If you already have one or more CloudWatch logs configured, click the **Edit** (pencil) icon to the right of **CloudWatch Logs(Optional)**.
3. In the **New or existing log group** box, type a log group name to organize CloudTrail events for you to review using CloudWatch Logs, and then click **Continue**.

#### Note

For recommended log group naming conventions, see [Log Group and Log Stream Names](#).

Next, specify a role for CloudTrail to assume to deliver events to the log stream.

### Specify a Role

#### To specify a role using the console

1. By default, the **CloudTrail\_CloudWatchLogs\_Role** is selected for you. To verify this, click **View Details**. In the **IAM Role** box, the **CloudTrail\_CloudWatchLogs\_Role** is selected. The default role policy contains the permissions required for creating a CloudWatch Logs log stream in a log group that you specify and for delivering CloudTrail events to that log stream. To see the contents of the role policy, click **View Policy Document**.

#### Note

You can specify another role, but you must attach the appropriate role policy to the existing role if you want to use it to send log events to CloudWatch Logs.

2. Click **Allow**.

When you are finished with these steps in the console, the CloudTrail trail will be set up to use the log group and role you specified to send events to CloudWatch Logs.

## Configuring CloudWatch Logs Monitoring Using the CLI

You can use the AWS CLI to configure CloudTrail to send log events to CloudWatch Logs for monitoring.

## Creating a Log Group

1. If you do not already have an existing log group, create a CloudWatch Logs log group as a delivery endpoint for log events using the CloudWatch Logs command `create-log-group`.

```
aws logs create-log-group --log-group-name name
```

For example:

```
aws logs create-log-group --log-group-name CloudTrail/logs
```

2. Retrieve the log group Amazon Resource Name (ARN).

```
aws logs describe-log-groups
```

## Create a Policy Document

When you use the AWS CLI to create a role and attach a role policy, you must create a policy document as a JSON file. Open a text editor. Type the following and then save the policy document with the `.json` file extension. The following grants CloudTrail the required permissions to create a CloudWatch Logs log stream in the log group that you specify and to deliver CloudTrail events to that log stream. (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-1: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-1:accountID:log-group:log_group_name:log-
stream:CloudTrail_log_stream_name_prefix*"
      ]
    }
  ]
}
```

## Creating a Role

Create a role for CloudTrail to use to send events to the CloudWatch Logs log group using the IAM `create-role` command.

```
aws iam create-role --role-name role_name --assume-role-policy-document  
file://policy_document.json
```

Running this command attaches a trust policy to the specified role so that CloudTrail can assume the role. Take note of the role ARN.

## 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 AWS CLI commands in these procedures, see the [AWS CloudTrail Command Line Reference](#).

## Limitation

Because CloudWatch Logs has an [event size limitation of 256KB](#), CloudTrail does not send events larger than 256KB to CloudWatch Logs.

# Using an AWS CloudFormation Template to Create CloudWatch Alarms

You can create CloudWatch metric filters and alarms that monitor the CloudTrail events that you specify and send you notifications when the events occur. You can create your filters and alarms separately, or by using an AWS CloudFormation template to define them all at once.

This topic describes an example CloudFormation template from AWS that you can use as is, or as a starting point or as a reference for creating your own templates. For information on creating CloudWatch metric filters and alarms individually, see [Creating CloudWatch Alarms for CloudTrail Events: Examples \(p. 104\)](#).

## The Example CloudFormation Template

The downloadable and editable example CloudFormation template from AWS contains predefined CloudWatch metric filters and alarms that enable you to receive email notifications when certain security-related API calls are made in your AWS account. You can download the template directly from the following link: [https://s3-us-west-2.amazonaws.com/awscloudtrail/cloudwatch-alarms-for-cloudtrail-api-activity/CloudWatch\\_Alarms\\_for\\_CloudTrail\\_API\\_Activity.json](https://s3-us-west-2.amazonaws.com/awscloudtrail/cloudwatch-alarms-for-cloudtrail-api-activity/CloudWatch_Alarms_for_CloudTrail_API_Activity.json).

The example template defines metric filters that monitor creation and deletion of, or updates to, security groups, network ACLs, internet gateways, Amazon EC2 instances, and IAM policies. For each filter, the template describes a corresponding alarm that enables you to receive email notifications when a call to one of the APIs being monitored by the filter is made.

By default, most of the filters in the template trigger an alarm when one monitored event occurs within a five-minute period. You can modify these alarm thresholds for your own requirements. For example, you could monitor for 3 events in a 10-minute period. To make the changes, you can edit the template directly

or, after following the steps in the section that follows ([Using the CloudFormation template \(p. 96\)](#)), you can alter the thresholds in the [CloudWatch console](#).

**Note**

Because CloudTrail typically delivers log files every five minutes, it is highly recommended that you specify alarm periods of five minutes or more.

For a description of each of the metric filters and alarms in the template, and the API calls for which email notifications are triggered, see the section [CloudFormation Template Contents \(p. 100\)](#) later in this document.

## Using the CloudFormation template

To use the template:

1. Configure CloudTrail log file delivery to CloudWatch Logs. See [Sending CloudTrail Events to CloudWatch Logs \(p. 92\)](#).

**Note**

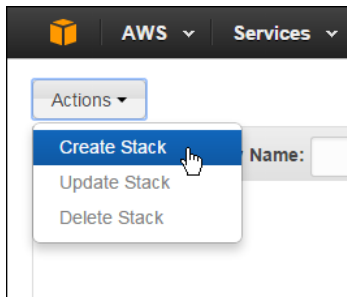
If you change the default log group name provided by CloudTrail, note it so that you can use it in the next step.

2. Create a AWS CloudFormation stack by using the template. A CloudFormation stack is a collection of related resources that you provision and update as a single unit.

The next section shows you how to create the stack and validate the email address that will receive any notifications that are generated.

### Create a CloudFormation Stack

1. Download the CloudFormation template from [https://s3-us-west-2.amazonaws.com/awsccloudtrail/cloudwatch-alarms-for-cloudtrail-api-activity/CloudWatch\\_Alarms\\_for\\_CloudTrail\\_API\\_Activity.json](https://s3-us-west-2.amazonaws.com/awsccloudtrail/cloudwatch-alarms-for-cloudtrail-api-activity/CloudWatch_Alarms_for_CloudTrail_API_Activity.json).
2. Go to the [CloudFormation console](#) and create a new stack.



3. On the **Select Template** page, give the stack a name. This example uses `CloudWatchAlarmsForCloudTrail`.

**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

---

### Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

---

#### Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

**Name**

4. Under **Template**, select **Upload a template to Amazon S3**.

### Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more](#).

---

#### Source

Select a sample template

Upload a template to Amazon S3  
 No file chosen

Specify an Amazon S3 template URL

5. Click **Choose File**, and then browse to and select the CloudFormation template that you downloaded.
6. Click **Next**.
7. On the **Specify Parameters** page, provide the email address that will receive notifications, and the enter name of the log group name that you used when you configured CloudTrail log file delivery to CloudWatch Logs.

**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

---

### Specify Parameters

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

#### Parameters

**Email**   
Enter the email address where you would like to receive notifications

**LogGroupName**   
Enter CloudWatch Logs log group name. Default is CloudTrail/DefaultLogGroup

[Cancel](#) [Previous](#) [Next](#)

8. Click **Next**.
9. On the **Options** page, you can create tags or configure other advanced options. These are not required.

### Options

#### Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 10 unique key-value pairs for each stack. [Learn more.](#)

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text"/>	<input type="text"/>	<a href="#">+</a>

► **Advanced**

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

[Cancel](#) [Previous](#) [Next](#)

10. Click **Next**.
11. On the **Review** page, verify that the template, email, log group, and other options, if any, are correct.

**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

### Review

---

#### Template

<b>Name</b>	CloudWatchAlarmsForCloudTrail
<b>Template URL</b>	<a href="https://s3-us-west-2.amazonaws.com/cf-templates-1kf4fpqkb57z-us-west-2/2015060P4H-CloudWatch_Alarms_for_CloudTrail_API_Activity.json">https://s3-us-west-2.amazonaws.com/cf-templates-1kf4fpqkb57z-us-west-2/2015060P4H-CloudWatch_Alarms_for_CloudTrail_API_Activity.json</a>
<b>Description</b>	AWS CloudTrail API Activity Alarm Template for CloudWatch Logs
<b>Estimate cost</b>	<a href="#">Cost</a>

---

#### Parameters

<b>Email</b>	
<b>LogGroupName</b>	CloudTrail/DefaultLogGroup
<b>Create IAM resources</b>	False

---

#### Options

#### Tags

No tags provided

#### Advanced

<b>Notification</b>	
<b>Timeout</b>	none
<b>Rollback on failure</b>	Yes

12. Click **Create**. The stack will be created in a few minutes.

	Stack Name	Created Time	Status	Description
<input type="checkbox"/>	CloudWatchAlarmsForCloudTrail	2015-02-28 17:18:55 UTC-0800	CREATE_COMPLETE	AWS CloudTrail API Activ

13. After the CloudFormation stack has been created, you will receive an email at the address that you specified to validate it.

14. In the email, click **Confirm subscription**.

You have chosen to subscribe to the topic:  
**arn:aws:sns:us-west-2:111222333444:CloudWatchAlarmsForCloudTrail-AlarmNotificationTopic-22ABC2DEFGHI2**


To confirm this subscription, click or visit the link below (If this was in error no action is necessary):  
[Confirm subscription](#)

Please do not reply directly to this e-mail. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send email to [sns-opt-out](#)

You will now receive email notifications when the alarms specified by the template are triggered.

## AWS CloudTrail User Guide

### Using an AWS CloudFormation Template to Create CloudWatch Alarms



Sat 2/28/2015 7:11 PM  
AWS Notifications <no-reply@sns.amazonaws.com>  
ALARM: "CloudTrailIAMPolicyChanges" in US-West-2

To

You are receiving this email because your Amazon CloudWatch Alarm "CloudTrailIAMPolicyChanges" in the US-West-2 region has entered the ALARM state, because "Threshold Crossed: 1 datapoint (7.0) was greater than or equal to the threshold (1.0)." at "Sunday 01 March, 2015 03:10:34 UTC".

View this alarm in the AWS Management Console:  
<https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#s=Alarms&alarm=CloudTrailIAMPolicyChanges>

Alarm Details:

- Name: CloudTrailIAMPolicyChanges
- Description: Alarms when an API call is made to change an IAM policy.
- State Change: INSUFFICIENT\_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint (7.0) was greater than or equal to the threshold (1.0).
- Timestamp: Sunday 01 March, 2015 03:10:34 UTC
- AWS Account: 111122223333

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for 300 seconds.

Monitored Metric:

- MetricNamespace: CloudTrailMetrics
- MetricName: IAMPolicyEventCount
- Dimensions:
- Period: 300 seconds
- Statistic: Sum
- Unit: not specified

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-west-2:111122223333:CWLAlarms1234-AlarmNotificationTopic-ABC1DEFG234H]
- INSUFFICIENT\_DATA:

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:  
<https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:111122223333:CWLAlarms1234-AlarmNotificationTopic-ABC1DEFG234H:78080d51-8221-4ff5-b4b5-0366898a7d0a&Endpoint=janedoe@amazon.com>

Please do not reply directly to this e-mail. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

You can review the metric filter or alarm definitions in the [CloudWatch console](#).

## CloudFormation Template Contents

The following tables show each of the metric filters and alarms in the template, their purpose, and the API calls for which email notifications are triggered. Notifications are triggered when one or more of the API calls for a listed filter are made.



**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

**Amazon S3 Bucket Events**

<b>Metric Filter and Alarm</b>	<b>Monitor and Send Notifications for</b>	<b>Notifications triggered by one or more of</b>
S3BucketChangesMetricFilter S3BucketChangesAlarm	API calls that change bucket policy, lifecycle, replication, or ACLs	PutBucketAcl PutBucketPolicy PutBucketCors PutBucketLifecycle PutBucketReplication DeleteBucketPolicy DeleteBucketCors DeleteBucketLifecycle DeleteBucketReplication

**Network Events**

<b>Metric Filter and Alarm</b>	<b>Monitor and Send Notifications for</b>	<b>Notifications triggered by one or more of</b>
SecurityGroupChangesMetricFilter SecurityGroupChangesAlarm	API calls that create, update and delete Security Groups	AuthorizeSecurityGroupIngress AuthorizeSecurityGroupEgress RevokeSecurityGroupIngress RevokeSecurityGroupEgress CreateSecurityGroup DeleteSecurityGroup
NetworkAclChangesMetricFilter NetworkAclChangesAlarm	API calls that create, update and delete Network ACLs	CreateNetworkAcl CreateNetworkAclEntry DeleteNetworkAcl DeleteNetworkAclEntry ReplaceNetworkAclAssociation ReplaceNetworkAclEntry

**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

<b>Metric Filter and Alarm</b>	<b>Monitor and Send Notifications for</b>	<b>Notifications triggered by one or more of</b>
GatewayChangesMetricFilter GatewayChangesAlarm	API calls that create, update and delete customer and Internet gateways	CreateCustomerGateway DeleteCustomerGateway AttachInternetGateway CreateInternetGateway DeleteInternetGateway DetachInternetGateway
VpcChangesMetricFilter VpcChangesAlarm	API calls that create, update and delete Virtual Private Clouds (VPCs), VPC peering connections and VPC connections to classic EC2 instances using ClassicLink	CreateVpc DeleteVpc ModifyVpcAttribute AcceptVpcPeeringConnection CreateVpcPeeringConnection DeleteVpcPeeringConnection RejectVpcPeeringConnection AttachClassicLinkVpc DetachClassicLinkVpc DisableVpcClassicLink EnableVpcClassicLink

**Amazon EC2 Events**

<b>Metric Filter and Alarm</b>	<b>Monitor and Send Notifications for</b>	<b>Notifications triggered by one or more of</b>
EC2InstanceChangesMetricFilter EC2InstanceChangesAlarm	The creation, termination, start, stop, and reboot of EC2 instances	RebootInstances RunInstances StartInstances StopInstances TerminateInstances

**AWS CloudTrail User Guide**  
**Using an AWS CloudFormation Template to Create**  
**CloudWatch Alarms**

Metric Filter and Alarm	Monitor and Send Notifications for	Notifications triggered by one or more of
EC2LargeInstanceChangesMetricFilter  EC2LargeInstanceChangesAlarm	The creation, termination, start, stop, and reboot of 4x and 8x large EC2 instances	<u>At least one of</u>  RebootInstances  RunInstances  StartInstances  StopInstances  TerminateInstances  <u>and at least one of:</u>  instancetype=*.4xlarge  instancetype=*.8xlarge

**CloudTrail and IAM Events**

Metric Filter and Alarm	Monitor and Send Notifications for	Notifications triggered by one or more of these calls (or activity)
CloudTrailChangesMetricFilter  CloudTrailChangesAlarm	The creation or deletion of trails, or updates to trails. The occurrence of start and stop logging events for a trail.	CreateTrail  DeleteTrail  StartLogging  StopLogging  UpdateTrail
ConsoleSignInFailuresMetricFilter  ConsoleSignInFailuresAlarm	Console login failures	eventName is ConsoleLogin <i>and</i> errorMessage is "Failed authentication"
AuthorizationFailuresMetricFilter  AuthorizationFailuresAlarm	Authorization failures	Any API call which results in an error code of AccessDenied <i>or</i> *UnauthorizedOperation.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

Metric Filter and Alarm	Monitor and Send Notifications for	Notifications triggered by one or more of these calls (or activity)
IAMPolicyChangesMetricFilter IAMPolicyChangesAlarm	Changes to IAM policies	DeleteGroupPolicy DeleteRolePolicy DeleteUserPolicy PutGroupPolicy PutRolePolicy PutUserPolicy CreatePolicy DeletePolicy CreatePolicyVersion DeletePolicyVersion AttachRolePolicy DetachRolePolicy AttachUserPolicy DetachUserPolicy AttachGroupPolicy DetachGroupPolicy

## Creating CloudWatch Alarms for CloudTrail Events: Examples

This topic describes how to configure alarms for CloudTrail events using example scenarios. Configuring alarms involves the following steps:

- Create a metric filter.
- Create an alarm.

**Note**

Instead of creating the metric filters and alarms that are presented here manually, you can use an AWS CloudFormation template to create them all at once. For more information, see [Using an AWS CloudFormation Template to Create CloudWatch Alarms \(p. 95\)](#).

**Topics**

- [Example: Amazon S3 Bucket Activity \(p. 105\)](#)
- [Example: Security Group Configuration Changes \(p. 107\)](#)
- [Example: Network Access Control List \(ACL\) Changes \(p. 109\)](#)
- [Example: Network Gateway Changes \(p. 111\)](#)

- [Example: Amazon Virtual Private Cloud \(VPC\) Changes \(p. 113\)](#)
- [Example: Amazon EC2 Instance Changes \(p. 115\)](#)
- [Example: EC2 Large Instance Changes \(p. 117\)](#)
- [Example: CloudTrail Changes \(p. 119\)](#)
- [Example: Console Sign-In Failures \(p. 121\)](#)
- [Example: Authorization Failures \(p. 123\)](#)
- [Example: IAM Policy Changes \(p. 125\)](#)

## Example: Amazon S3 Bucket Activity

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an Amazon S3 API call is made to `PUT` or `DELETE` bucket policy, bucket lifecycle, bucket replication, or to `PUT` a bucket ACL. The alarm also is triggered for the CORS (cross-origin resource sharing) `PUT` bucket and `DELETE` bucket events. For information about cross-origin resource sharing, see [Cross-Origin Resource Sharing](#).

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventSource = s3.amazonaws.com) && (($eventName = PutBucketAcl) ||
($eventName = PutBucketPolicy) || ($eventName = PutBucketCors) ||
($eventName = PutBucketLifecycle) || ($eventName = PutBucketReplication)
|| ($eventName = DeleteBucketPolicy) || ($eventName = DeleteBucketCors)
|| ($eventName = DeleteBucketLifecycle) || ($eventName = DeleteBucketRep
lication)) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **S3BucketActivity**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **S3BucketActivityEventCount**.
9. Click **Metric Value**, and then type **1**. If **Metric Value** does not appear, click **Advanced** first.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the **S3BucketActivity** filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

## Create Alarm

1. [Select Metric](#)    2. **Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

Whenever: S3BucketActivityEventCount

**2** is:

**3** for:  consecutive period(s)

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

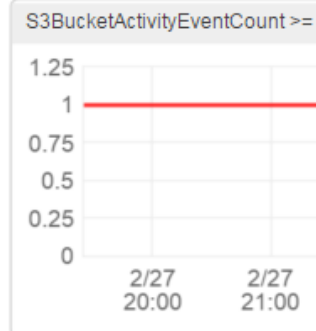
**6** Send notification to:  Select list i

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

### Alarm Preview

This alarm will trigger when the metric value is to or above the red line for a duration of 5 minutes.



Namespace: CloudTrailM

Metric Name:

**4** Period:




**5** Statistic:

[Cancel](#)

[Back](#)

Setting	Value
<b>1</b>	S3 Bucket Activity
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

Setting	Value
	Sum
	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

- When you are finished, click **Create Alarm**.

## Example: Security Group Configuration Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when any configuration changes happen involving security groups.

### Create a Metric Filter

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, click **Logs**.
- In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
- Click **Create Metric Filter**.
- On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = AuthorizeSecurityGroupIngress) || ($.eventName = AuthorizeSecurityGroupEgress) || ($.eventName = RevokeSecurityGroupIngress) || ($.eventName = RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) || ($.eventName = DeleteSecurityGroup) }
```

#### Note

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 User Guide.

- Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **SecurityGroupEvents**
- Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
- In the **Metric Name** field, enter **SecurityGroupEventCount**.
- Click **Metric Value**, and then type **1**.
- When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

- On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
- On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. **Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: SecurityGroupEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics

Metric Name:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	Security Group Configuration Changes
<b>2</b>	>=1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.



## Example: Network Access Control List (ACL) Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when any configuration changes happen involving network ACLs.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = CreateNetworkAcl) || ($.eventName = CreateNetworkAclEntry)
  || ($.eventName = DeleteNetworkAcl) || ($.eventName = DeleteNetworkAclEntry)
  || ($.eventName = ReplaceNetworkAclEntry) || ($.eventName = ReplaceNetwork
AclAssociation) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **NetworkACLEvents**.
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** box, enter **NetworkACLEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. **Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: NetworkACLEventCount

**2** is: >=

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics

Metric Name:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm: State is ALARM

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	Network ACL Configuration Changes
<b>2</b>	>=1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: Network Gateway Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an API call is made to create, update, or delete a customer or Internet gateway.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = CreateCustomerGateway) || ($.eventName = DeleteCustomerGateway) || ($.eventName = AttachInternetGateway) || ($.eventName = CreateInternetGateway) || ($.eventName = DeleteInternetGateway) || ($.eventName = DetachInternetGateway) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **GatewayChanges**.
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **GatewayEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Example: Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. **Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: GatewayEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics  
Metric Name: GatewayEventCount

**4** Period:

**5** Statistic:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel Back Next Create Alarm

Setting	Value
<b>1</b>	Network Gateway Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: Amazon Virtual Private Cloud (VPC) Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an API call is made to create, update or delete an Amazon VPC, an Amazon VPC peering connection or an Amazon VPC connection to classic Amazon EC2 instances.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = CreateVpc) || ($.eventName = DeleteVpc) || ($.eventName = ModifyVpcAttribute) || ($.eventName = AcceptVpcPeeringConnection) || ($.eventName = CreateVpcPeeringConnection) || ($.eventName = DeleteVpcPeeringConnection) || ($.eventName = RejectVpcPeeringConnection) || ($.eventName = AttachClassicLinkVpc) || ($.eventName = DetachClassicLinkVpc) || ($.eventName = DisableVpcClassicLink) || ($.eventName = EnableVpcClassicLink) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **VpcChanges**.
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **VpcEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. [Define Alarm](#)

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: VpcEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics

Metric Name:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	VPC Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: Amazon EC2 Instance Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an API call is made to create, terminate, start, stop or reboot an Amazon EC2 instance.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = RunInstances) || ($.eventName = RebootInstances) ||  
  ($.eventName = StartInstances) || ($.eventName = StopInstances) ||  
  ($.eventName = TerminateInstances) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **EC2InstanceChanges**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **EC2InstanceEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. [Define Alarm](#)

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: EC2InstanceEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics  
Metric Name: EC2InstanceEventCount

**4** Period:

**5** Statistic:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel Back Next Create Alarm

Setting	Value
<b>1</b>	EC2 Instance Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.



## Example: EC2 Large Instance Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an API call is made to create, terminate, start, stop or reboot a 4x or 8x-large EC2 instance.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ (($.eventName = RunInstances) || ($.eventName = RebootInstances) ||  
 ($.eventName = StartInstances) || ($.eventName = StopInstances) ||  
 ($.eventName = TerminateInstances)) && (($.requestParameters.instanceType  
 = *.8xlarge) || ($.requestParameters.instanceType = *.4xlarge)) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **EC2LargeInstanceChanges**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **EC2LargeInstanceEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. [Define Alarm](#)

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: EC2LargeInstanceEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics  
Metric Name:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	EC2 Large Instance Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: CloudTrail Changes

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an API call is made to create, update or delete a CloudTrail trail, or to start or stop logging to a trail.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = CreateTrail) || ($.eventName = UpdateTrail) || ($.eventName = DeleteTrail) || ($.eventName = StartLogging) || ($.eventName = StopLogging) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **CloudTrailChanges**.
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **CloudTrailEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. Select Metric
2. Define Alarm

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: CloudTrailEventCount

**2** is: >= 1

**3** for: 1 consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics  
Metric Name: CloudTrailEventCount

**4** Period: 5 Minutes

**5** Statistic: Sum

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm: State is ALARM

**6** Send notification to: NotifyMe Select list ⓘ

**7** Email list: user1@example.net.

+ Notification
+ AutoScaling Action
+ EC2 Action

Cancel Back Next Create Alarm

Setting	Value
<b>1</b>	CloudTrail Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: Console Sign-In Failures

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when there are three or more sign-in failures during a five minute period.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed authentication")
}
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **ConsoleSignInFailures**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** box, enter **ConsoleSigninFailureCount**.
9. Click **Show advanced metric settings**.
10. Click **Metric Value**, and then type **1**.
11. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. Select Metric
2. Define Alarm

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: ConsoleSigninFailureCount

**2** is: >= 3

**3** for: 1 consecutive period(s)

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm: State is ALARM

**6** Send notification to: NotifyMe Select list ⓘ

**7** Email list: user1@example.net.

+ Notification
+ AutoScaling Action
+ EC2 Action

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

ConsoleSigninFailureCount >= 3

Namespace: CloudTrailMetrics

Metric Name: ConsoleSigninFailureCc

**4** Period: 5 Minutes

**5** Statistic: Sum

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	Console Sign-in Failures
<b>2</b>	>=3
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## Example: Authorization Failures

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when an unauthorized API call is made.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.errorCode = "*UnauthorizedOperation") || ($.errorCode = "AccessDenied*")
}
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **AuthorizationFailures**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **AuthorizationFailureCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. Select Metric
2. Define Alarm

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: AuthorizationFailureCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics

Metric Name: AuthorizationFailureCou

**4** Period:

**5** Statistic:

### Actions

Define what actions are taken when your alarm changes state.

Notification
Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	Authorization Failures
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.



## Example: IAM Policy Changes

This scenario walks you through how to use the AWS Management Console 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.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ ($.eventName=DeleteGroupPolicy) || ($.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) }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **IAMPolicyChanges**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **IAMPolicyEventCount**.
9. Click **Metric Value**, and then type **1**.
10. When you are finished, click **Create Filter**.

### Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Examples**

**Create Alarm**
✕

1. [Select Metric](#)
2. **Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

---

Whenever: IAMPolicyEventCount

**2** is:

**3** for:  consecutive period(s)

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: CloudTrailMetrics

Metric Name:

### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

Cancel
Back
Next
Create Alarm

Setting	Value
<b>1</b>	IAM Policy Changes
<b>2</b>	1
<b>3</b>	1
<b>4</b>	5 Minutes
<b>5</b>	Sum
<b>6</b>	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
<b>7</b>	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

# Creating CloudWatch Alarms for CloudTrail Events: Additional Examples

[AWS Identity and Access Management \(IAM\) best practices](#) recommend that you do not use your root account credentials to access AWS. Instead, you should [create individual IAM users](#) so that you can give each user a unique set of security credentials. The IAM Best Practices also recommend that you [enable multi-factor authentication \(MFA\)](#) for IAM users who are allowed access to sensitive resources or APIs.

You can monitor whether activity in your AWS account adheres to these best practices by creating the CloudWatch alarms that notify you when root account credentials have been used to access AWS, or when API activity or console sign-ins without MFA have occurred. These alarms are described in this document.

Configuring an alarm involves two main steps:

- Create a metric filter
- Create an alarm based on the filter

## Topics

- [Example: Monitor for Root Usage \(p. 127\)](#)
- [Example: Monitor for API Activity Without Multi-factor Authentication \(MFA\) \(p. 129\)](#)
- [Example: Monitor for Console Sign In Without Multi-factor Authentication \(MFA\) \(p. 131\)](#)

## Example: Monitor for Root Usage

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when root (account) credentials are used.

### Create a Metric Filter

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, click **Logs**.
3. In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
4. Click **Create Metric Filter**.
5. On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ $.userIdentity.type = "Root" && $.userIdentity.invokedBy NOT EXISTS }
```

#### Note

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 User Guide.

6. Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **RootAccountUsage**
7. Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
8. In the **Metric Name** field, enter **RootAccountUsageCount**.
9. Click **Metric Value**, and then type **1**.

**Note**

If **Metric Value** does not appear, click **Show advanced metric settings** first.

10. When you are finished, click **Create Filter**.

## Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

### Create Alarm

1. Select Metric    **2. Define Alarm**

#### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

Whenever: RootAccountUsageCount

**2** is:

**3** for:  consecutive period(s)

#### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

[+ Notification](#)   [+ AutoScaling Action](#)   [+ EC2 Action](#)

#### Alarm Preview

This alarm will trigger when the b to or above the red line for a dura

RootAccountUsageCount >

1.25  
1  
0.75  
0.5  
0.25  
0

2/27 19:00    2/27 20:00    2/27 21:00

Namespace: CloudTrail

Metric Name:

**4** Period:

**5** Statistic:

[Cancel](#)   [Back](#)   [N](#)

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Additional Examples**

Setting	Value
1	Root Account Usage
2	>=1
3	1
4	5 Minutes
5	Sum
6	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
7	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

- When you are finished, click **Create Alarm**.

## Example: Monitor for API Activity Without Multi-factor Authentication (MFA)

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when API calls are made without the use of multi-factor authentication (MFA).

### Create a Metric Filter

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, click **Logs**.
- In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
- Click **Create Metric Filter**.
- On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ $.userIdentity.sessionContext.attributes.mfaAuthenticated != "true" }
```

#### Note

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 User Guide.

- Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **ApiActivityWithoutMFA**.
- Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
- In the **Metric Name** box, enter **ApiActivityWithoutMFACount**.
- Click **Metric Value**, and then type **1**.

**Note**

If **Metric Value** does not appear, click **Show advanced metric settings** first.

10. When you are finished, click **Create Filter**.

## Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

**Create Alarm**

1. Select Metric    **2. Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** Name:

Description:

Whenever: ApiActivityWithoutMFACount

**2** is:

**3** for:  consecutive period(s)

### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm:

**6** Send notification to:  [Select list](#) ⓘ

**7** Email list:

[+ Notification](#)   [+ AutoScaling Action](#)   [+ EC2 Action](#)

### Alarm Preview

This alarm will trigger when the b to or above the red line for a dura

ApiActivityWithoutMFACoun

Namespace: CloudTrail

Metric Name:

**4** Period:

**5** Statistic:

[Cancel](#)   [Back](#)   [Next](#)

**AWS CloudTrail User Guide**  
**Creating CloudWatch Alarms for CloudTrail Events:**  
**Additional Examples**

Setting	Value
1	Api Activity Without MFA
2	>=1
3	1
4	5 Minutes
5	Sum
6	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
7	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

- When you are finished, click **Create Alarm**.

## Example: Monitor for Console Sign In Without Multi-factor Authentication (MFA)

This scenario walks you through how to use the AWS Management Console to create an Amazon CloudWatch alarm that is triggered when a console sign in is made without multi-factor authentication.

### Create a Metric Filter

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, click **Logs**.
- In the list of log groups, select the check box next to the log group that you created for CloudTrail log events.
- Click **Create Metric Filter**.
- On the **Define Logs Metric Filter** screen, click **Filter Pattern** and then type the following:

```
{ $.eventName = "ConsoleLogin" && $.additionalEventData.MFAUsed != "No" }
```

#### Note

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 User Guide.

- Click **Assign Metric**, and then on the **Create Metric Filter and Assign a Metric** screen, in the **Filter Name** box, enter **ConsoleSignInWithoutMfa**
- Under **Metric Details**, in the **Metric Namespace** box, enter **CloudTrailMetrics**.
- In the **Metric Name** field, enter **ConsoleSignInWithoutMfaCount**.
- Click **Metric Value**, and then type 1.

#### Note

If **Metric Value** does not appear, click **Show advanced metric settings** first.

10. When you are finished, click **Create Filter**.

## Example: Create an Alarm

These steps are a continuation of the previous steps for creating a metric filter.

1. On the **Filters for *Log\_Group\_Name*** page, next to the filter name, click **Create Alarm**.
2. On the **Create Alarm** page, provide the following values.

### Create Alarm

[1. Select Metric](#) [2. Define Alarm](#)

#### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**1** **Name:**

**2** **Description:**

**Whenever:** ConsoleSignInWithoutMfaCount

**2** **is:**

**3** **for:**  consecutive period(s)

#### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

**Whenever this alarm:**

**6** **Send notification to:**  [Select list](#) **i**

**7** **Email list:**

[+ Notification](#) [+ AutoScaling Action](#) [+ EC2 Action](#)

#### Alarm Preview

This alarm will trigger when the metric value is at or above the red line for a duration of 5 minutes.

**Namespace:** CloudTrail








**Metric Name:**

**4** **Period:**

**5** **Statistic:**

[Cancel](#) [Back](#) [Next](#)



Setting	Value
	Console Sign In Without MFA
	1
	1
	5 Minutes
	Sum
	Near the <b>Select a notification list</b> box, click <b>New list</b> , and then type a unique topic name for the list.
	Click <b>Email list</b> , and then type the email address to which you want notifications sent. (You will receive an email at this address to confirm that you created this alarm.)

3. When you are finished, click **Create Alarm**.

## 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 events to CloudWatch Logs by deleting the delivery endpoint.

### AWS Management Console

#### To remove the CloudWatch Logs delivery endpoint using the AWS Management Console

1. Sign in to the AWS Management Console.
2. Navigate to the CloudTrail console.
3. In the navigation pane, click **Configuration**.
4. In the **CloudWatch Logs (optional)** section, click the **Delete** (trash can) icon.
5. Click **Continue** to confirm.

## AWS Command Line Interface (CLI)

You can remove the CloudWatch Logs log group as a delivery endpoint using the `update-trail` command. The following command clears the log group and role from the trail configuration.

```
aws cloudtrail update-trail --name trailname --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`. Log group names can be between 1 and 512 characters long. Allowed characters include a-z, A-Z, 0-9, '\_' (underscore), '-' (hyphen), '/' (forward slash), and '.' (period).

When CloudTrail creates the log stream for the log group, it names the log stream according to the following format: `account_ID_CloudTrail_source_region`.

## Role Policy Document for CloudTrail to Use CloudWatch Logs for Monitoring

This section describes the trust 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 CloudTrail Events to CloudWatch Logs](#) (p. 92). You can also create a role using IAM. For more information, see [Creating a Role for an AWS Service \(AWS Management Console\)](#) or [Creating a Role \(CLI and API\)](#).

The following 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. (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-1: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-1:accountID:log-group:log_group_name:log-  
stream:CloudTrail_log_stream_name_prefix*" ]  
    }  
  ]  
}
```

```
        "logs:PutLogEvents"  
    ],  
    "Resource": [  
        "arn:aws:logs:us-east-1:accountID:log-group:log_group_name:log-  
stream:CloudTrail_log_stream_name_prefix*"  
    ]  
  }  
]
```

## Using the CloudTrail Processing Library

The CloudTrail Processing Library is a Java library that provides an easy way to process AWS CloudTrail logs in a fault-tolerant, scalable and flexible way. You provide configuration details about your CloudTrail SQS queue and write code to process events. The CloudTrail Processing Library does the rest, polling your Amazon SQS queue, reading and parsing queue messages, downloading CloudTrail log files, parsing events in the log files and passing them to your code as Java objects. The CloudTrail Processing Library is highly scalable and fault-tolerant, handling parallel processing of log files so that you can process as many logs as necessary, and robustly handling network failures related to network timeouts and inaccessible resources.

This chapter provides information about 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 \(p. 135\)](#)
- [Processing CloudTrail Logs with the CloudTrail Processing Library \(p. 135\)](#)
- [Advanced Topics \(p. 139\)](#)
- [Additional Resources \(p. 142\)](#)

## Minimum requirements

To use the CloudTrail Processing Library, you must have the following:

- [AWS SDK for Java 1.9.3](#)
- [Java 1.7](#)

## Processing CloudTrail Logs with the CloudTrail Processing Library

To use the CloudTrail Processing Library to process CloudTrail logs in your Java application:

1. [Add the CloudTrail Processing Library to your Project \(p. 136\)](#)
2. [Configure the CloudTrail Processing Library \(p. 137\)](#)

3. [Implement the Events Processor \(p. 138\)](#)
4. [Instantiate and Run the Processing Executor \(p. 139\)](#)

## Add the CloudTrail Processing Library to your Project

To use the CloudTrail Processing Library you must add it to your Java project's classpath.

### Topics

- [Adding the Library to an Apache Ant Project \(p. 136\)](#)
- [Adding the Library to an Apache Maven Project \(p. 136\)](#)
- [Adding the CloudTrail Processing Library to an Eclipse Project \(p. 137\)](#)

### Adding the Library to an Apache Ant Project

#### To add the CloudTrail Processing Library to an Ant project

1. Download or clone the CloudTrail Processing Library source code from GitHub at:
  - <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-1.0.0.jar"/>
</classpath>
```

### Adding the Library to an Apache Maven Project

The CloudTrail Processing Library is available for [Apache Maven](#), so adding it to your project is as easy as writing a single dependency in your project's `pom.xml` file.

#### To add the CloudTrail Processing Library to a Maven project

- Using your favorite text editor, 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>1.0.0</version>
</dependency>
```

## Adding the CloudTrail Processing 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 at:

- <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-1.0.0.jar to a directory in your project (typically lib).
4. Right-click your project's name in the Eclipse **Project Explorer**, and select **Build Path > Configure**
5. In the **Java Build Path** window, click the **Libraries** tab.
6. Click **Add JARs...** and navigate to the path where you copied aws-cloudtrail-processing-library-1.0.0.jar.
7. Click **OK** to complete adding the .jar to your project.

## Configure 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. Here is an example file that demonstrates 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
```

The required parameters are *sqsUrl*, *accessKey* and *secretKey*. The *sqsUrl* parameter provides the URL to pull your CloudTrail notifications from. If you don't provide this value, then an `IllegalStateException` will be thrown by the `AWSCloudTrailProcessingExecutor`. The *accessKey* and *secretKey* parameters provide your AWS credentials to the library, allowing it to access AWS on your behalf.

The other parameters have reasonable defaults that are set by the library. For information about the default values for each option, see the [AWS 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.

For 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);
```

## Implement the Events Processor

To process CloudTrail logs, you must implement a `EventsProcessor` that receives the CloudTrail log data. Here 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 a `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 just prints the event information for each event passed to `SampleEventsProcessor`. In your own implementation, you can process logs as you see fit. The `AWSCloudTrailProcessingExecutor` will continue to send events to your `EventsProcessor` as long as it has events to send and is still running.

## Instantiate and Run the Processing Executor

Once you have written a `EventsProcessor` and have 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 a `AWSCloudTrailProcessingExecutor`.

### To use `AWSCloudTrailProcessingExecutor` to process CloudTrail events

1. Instantiate an `AWSCloudTrailProcessingExecutor.Builder` object. `Builder`'s constructor takes a `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 \(p. 139\)](#)
- [Reporting Progress \(p. 141\)](#)
- [Handling Errors \(p. 142\)](#)

## Filtering the Events to Process

By default, all of the logs in your Amazon SQS queue's S3 bucket and all of the events that they contain will be 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 or not 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 `filterSource()` method is called by the CloudTrail Processing Library after the library has polled for logs on the Amazon SQS queue, but before event filtering or processing has been done for those logs.

Here 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` will be 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 will be sent to your `EventsProcessor` or not. `EventFilter` declares a single callback method, `filterEvent()`, that receives a `CloudTrailEvent` object. To keep the event from being processed, return `false` from `filterEvent()`.

The `filterEvent()` method is called by the CloudTrail Processing Library after the library has polled for logs on the Amazon SQS queue and after source filtering, but before event processing has been done for those logs.

Here is an 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` will be used, which allows all events to be processed (it always returns `true`).

## Reporting Progress

The `ProgressReporter` interface can be implemented 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 being performed (in the `progressState` member, which holds a member of the `ProgressState` enumeration that identifies the current operation) and can contain additional information (in the `progressInfo` member). Additionally, any object that you return from `reportStart()` will be passed to `reportEnd()`, so you can provide contextual information such as what time it was when the event began processing.

Here 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 =  
        LoggerFactory.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, will be 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 being 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, as well.

Here is an example implementation:

```
public class SampleExceptionHandler implements ExceptionHandler{
    private static final Log logger =
        LoggerFactory.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 simply prints a standard error message, will be used instead.

## Additional Resources

For more information about the CloudTrail Processing Library, see the following additional resources:

- The [CloudTrail Processing Library](#) GitHub project, which includes [sample](#) code that demonstrates how to implement a CloudTrail Processing Library application
- The [CloudTrail Processing Library Java Package Documentation](#)

# CloudTrail Event Reference

A CloudTrail log is a record in JSON format that contains information about requests for resources in your account—what service was accessed, what action was performed, and any parameters for the action. The request also helps you determine who made the request. The event data is enclosed in a `Records` array.

The following example shows a single log record at the beginning of a log file. The entry indicates that an IAM user named Alice called the CloudTrail **StartLogging** API by using the CloudTrail console to start the logging process.

```
{
  "Records": [{
    "eventVersion": "1.01",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIDAJDPLRKL7UEXAMPLE",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "Alice",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2014-03-18T14:29:23Z"
        }
      }
    },
    "eventTime": "2014-03-18T14:30:07Z",
    "eventSource": "cloudtrail.amazonaws.com",
    "eventName": "StartLogging",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "72.21.198.64",
    "userAgent": "AWSConsole, aws-sdk-java/1.4.5 Linux/x.xx.fleetxen Java_Hot
Spot(TM)_64-Bit_Server_VM/xx",
    "requestParameters": {
      "name": "Default"
    },
    "responseElements": null,
  }],
}
```

```
"requestID": "cdc73f9d-aea9-11e3-9d5a-835b769c0d9c",  
"eventID": "3074414d-c626-42aa-984b-68ff152d6ab7"  
},  
  ... additional entries ...  
]
```

The following topics list the fields that CloudTrail uses to represent the data it captures for each AWS API call and sign-in event.

#### Topics

- [CloudTrail Record Contents \(p. 144\)](#)
- [CloudTrail userIdentity Element \(p. 146\)](#)
- [Tracing Federated Identities in CloudTrail Logs \(p. 149\)](#)
- [Non-API Events Captured by CloudTrail \(p. 150\)](#)

## CloudTrail Record Contents

The body of the record contains fields that you can use to determine what action was requested and when and where the request was made.

#### **eventTime**

The date and time the request was made, in coordinated universal time (UTC).

#### **eventVersion**

The version of the log event format. The current version is 1.02.

#### **userIdentity**

Information about the user that made a request. For more information, see [CloudTrail userIdentity Element \(p. 146\)](#)

#### **eventSource**

The service that the request was made to. This name is normally a short form of the service name without spaces plus `.amazonaws.com`. For example, the `eventSource` field lists a call to AWS CloudFormation as `cloudformation.amazonaws.com`, a call to Amazon EC2 as `ec2.amazonaws.com`, and a call to Amazon Simple Workflow Service as `swf.amazonaws.com`. One exception to this convention is CloudWatch, for which the event source is `monitoring.amazonaws.com`.

#### **eventName**

The requested action, which is one of the actions listed in the API Reference for the service.

#### **awsRegion**

The AWS region that the request was made to; for example, `us-east-1`. For a list of the region codes for the regions supported by CloudTrail, see [CloudTrail Supported Regions \(p. 13\)](#).

#### **sourceIPAddress**

The apparent 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 AWS, only the DNS name is displayed.

### **userAgent**

The agent through which the request was made, such as the AWS Management Console or an AWS SDK. For example:

- `aws-cli/1.3.23 Python/2.7.6 Linux/2.6.18-164.el5`. The request was made using the AWS CLI installed on Linux.
- `AWSConsole`. The request was made through the AWS Management Console.
- `aws-sdk-java`. The request was made through the AWS SDK for Java.
- `aws-sdk-ruby`. The request was made through the AWS SDK for Ruby.

#### **Note**

For events originated by AWS, this field will generally be of the form `aws-internal/#` where `#` is a number used for internal purposes.

### **errorCode**

The AWS service error if the request returns an error.

### **errorMessage**

If the request returns an error, the description of the error. This message includes messages for authorization failures. For such messages, CloudTrail captures the message logged by the service in its exception handling.

#### **Note**

Some AWS services provide the `errorCode` and `errorMessage` as top-level fields in the event. Other AWS services include error information as part of `responseElements`.

### **requestParameters**

The parameters, if any, that were sent with the request. These are fully documented in the API Reference documentation for the appropriate AWS service.

### **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 are fully documented in the API Reference documentation for the appropriate AWS service.

### **requestID**

Value generated by the service being called that identifies the request.

Support for this field begins with **eventVersion** 1.01.

### **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.

Support for this field begins with **eventVersion** 1.01.

### **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` - The service generated an event relevant to your trail. For example, this could occur when another account made a call with a resource that you own.

- **ConsoleSignin** - A user in your account (root, IAM, Federated, SAML, or SwitchRole) signed into the AWS Management Console.  
Support for these fields begins with **eventVersion** 1.02.

#### **apiVersion**

Identifies the API version number associated with the `AwsApiCall` **eventType** value.

Support for this field begins with **eventVersion** 1.02.

#### **recipientAccountID**

Represents the account ID that received this event. The **recipientAccountID** may be different from the [CloudTrail userIdentity Element \(p. 146\)](#) **accountId**. This can occur in cross-account resource access. For example, if a 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 different for the event delivered to the account which owns the KMS key.

Support for this field begins with **eventVersion** 1.02.

## CloudTrail userIdentity Element

AWS Identity and Access Management (IAM) provides different types of identities. The `userIdentity` element contains details that tell you what type of IAM identity made the request, what credentials were used, and if temporary, how the credentials were obtained.

### Examples

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": "AKIAIOSFODNN7EXAMPLE",
  "userName": "Alice"
}
```

The following example shows a `userIdentity` element for a request made using temporary security credentials obtained by assuming an IAM role. In this case, the element contains additional details about the role that was assumed to get credentials.

```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROAI DPPEZS35WEXAMPLE:AssumedRoleSessionName",
  "arn": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/MySessionName",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "attributes": {
      "creationDate": " 20131102T010628Z  ",

```

```

    "mfaAuthenticated": "false"
  },
  "sessionIssuer": {
    "type": "Role",
    "principalId": "AROAI DPPEZS35WEXAMPLE",
    "arn": "arn:aws:iam::123456789012:role/RoleToBeAssumed",
    "accountId": "123456789012",
    "userName": "RoleToBeAssumed"
  }
}

```

## 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 using your AWS account credentials. If the `userIdentity` type is `root` and you have set up an alias for your account, the `userName` field contains your account alias. For more information about account aliases, see [Your AWS Account ID and Its Alias](#).
- `IAMUser` – The request was made using the credentials of an IAM user.
- `AssumedRole` – The request was made with temporary security credentials that were obtained via a role via a call to the AWS STS `AssumeRole` API. This can include [roles for Amazon EC2](#) and [cross-account API access](#).
- `FederatedUser` – The request was made with temporary security credentials that were obtained via a call to the AWS STS `GetFederationToken` API. The `sessionIssuer` element indicates whether the API was called using root or IAM user credentials.

For more information about temporary security credentials, see [Using Temporary Security Credentials](#).

### 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 have not set up an alias for your AWS account, the <code>userName</code> field does not appear. For more information about account aliases, see <a href="#">Your AWS Account ID and Its Alias</a> . Note that the <code>userName</code> field will never contain Root because root is an identity type, not a user name.
Root (alias set)	The account alias	For more information about AWS account aliases, see <a href="#">Your AWS Account ID and Its Alias</a> .
IAMUser	The user name of the IAM user	
Assumed-Role	The name of the role assumed by the IAM user	
Federated-User	Not present	The <code>sessionContextsessionIssuer</code> section contains information about the identity that issued the session for the federated user. For more information, see <code>sessionIssuer</code> later in this topic.

### 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 AWS 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.

### 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.

### accountId

The account that owns the entity that granted permissions for the request. If the request was made using temporary security credentials, this is the account that owns the IAM user or role that was used to obtain credentials.

### accessKeyId

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

### sessionContext

If the request was made with temporary security credentials, an element that provides information about the session that was created for those credentials. Sessions are created when any API is called that returns temporary credentials. Sessions are also created when users work in the console and when users make a request using APIs that include [multi-factor authentication](#). Attributes for this element are:

- **creationDate** – The date and time when the temporary security credentials were issued. Represented in ISO 8601 basic notation.
- **mfaAuthenticated** – `true` if the root user or IAM user whose credentials were used for the request also was authenticated using an MFA device; otherwise, `false`.

### invokedBy

If the request was made by another AWS service, such as Auto Scaling or AWS Elastic Beanstalk, the name of the service.

### sessionIssuer

If the request was made with temporary security credentials, an element that provides information about how the credentials were obtained. For example, if the temporary security credentials were obtained by assuming a role, this element provides information about the assumed role. If the credentials were obtained by using root or IAM user credentials to call AWS STS `GetFederationToken`, the element provides information about the root account or IAM user. Attributes for this element are:

- **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 <b>userName</b> field does not appear. For more information about AWS account aliases, see <a href="#">Your AWS Account ID and Its Alias</a> . Note that the <b>userName</b> field will never contain <code>Root</code> because root is an identity type, not a user name.
Root (alias set)	The account alias	For more information about AWS account aliases, see <a href="#">Your AWS Account ID and Its Alias</a> .
IAMUser	The user name of the IAM user	This also applies when a federated user is using a session issued by <code>IAMUser</code> .
Role	The role name	Can be the role assumed by an IAM user, AWS service, or Web Identity Federated user in a role session.

- **principalId** – The internal ID of the entity that was 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.

#### **webIdFederationData**

If the request was made with temporary security credentials obtained using [web identity federation](#), an element that lists information about the identity provider. Attributes for this element are:

- **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. For more information, see [Available Keys for Web Identity Federation](#) in the *IAM User Guide* guide.

## Tracing Federated Identities in CloudTrail Logs

AWS IAM provides you with the ability to extend your organization's on-premises identities by using federation mechanisms in IAM and AWS STS. If you configure your federated identity solution with the same identities on-premises and in AWS, you can use CloudTrail logs to trace the identity of a user who made a particular API call. To enable this, you can create an identity broker application that gives your organization's LDAP or Active Directory (AD) users seamless access to AWS resources.

### Characteristics of an Identity Broker Application

An identity broker application that connects your organization's users to AWS has the following characteristics:

- The identity broker application has permissions to create temporary security credentials by calling the [AWS Security Token Service API](#).
- The identity broker application is able to verify that employees are authenticated within your existing authentication system.

- The identity broker application can provide users with a temporary URL that gives them access to the AWS Management Console without having to provide their credentials again (this is referred to as single sign-on).

For more information, see [Using Your Organization's Authentication System to Grant Access to AWS Resources](#).

## How an Identity Broker Application Works

The following is a summary of the steps that an identity broker application takes to seamlessly give a user in your organization's directory access to AWS.

1. The identity broker application calls `GetFederationToken` on behalf of an on-premises user with the `Name`, `Policy`, and `DurationSeconds` parameters. The string that the application specifies in the `Name` parameter becomes the federated user name.

### Important

To make tracing easier in CloudTrail logs, this name should be the same as the on-premises user name, as explained in the next section.

2. The identity broker application gets back a temporary set of credentials.
3. The application returns the temporary set of credentials to the user.
4. The user makes all subsequent API calls (like `RunInstances`) to AWS using these the credentials.
5. CloudTrail records the caller of the API as being the federated user that was specified in the `Name` parameter in the call to `GetFederationToken` in step 1.

## Enabling Your Identity Broker Application to Trace Federated AWS Users

To use CloudTrail logs to trace federated AWS users back to their corresponding on-premises identities, your identity broker application should keep the on-premises and AWS identities the same. You implement this by passing each of your organization's on-premises user names without change to the `Name` parameter of the `GetFederationToken` API call (step 1 in the previous workflow).

When you do so, all subsequent calls made by the federated user to AWS resources (like Amazon EC2 instances, DynamoDB, or Amazon S3 buckets) will contain the name of the federated user. The name that was passed to `GetFederationToken` is the name that CloudTrail records as the caller to the AWS API.

In CloudTrail logs, the identity of the federated user that made the call appears at the end of the Amazon Resource Name (ARN), as in the following example:

```
arn:aws:sts::123456789012:federated-user/Bob
```

If you do not configure your identity broker application to ensure that your on-premises directory and AWS federated user names are the same, you may find it impossible or difficult to use CloudTrail logs to trace the API activity of your on-premises users when they access AWS.

## Non-API Events Captured by CloudTrail

In addition to logging AWS API calls, CloudTrail also captures other related events that might have a security or compliance impact on your AWS account or that might help you troubleshoot operational problems.

## Topics

- [AWS Console Sign-in Events \(p. 151\)](#)

# AWS Console Sign-in Events

CloudTrail records attempts to sign into the AWS Management Console, the AWS Discussion Forums and the AWS Support Center. All IAM user sign-in attempts (successes and failures), all federated user sign-in events (successes and failures) and all successful AWS root account sign-in attempts generate records in CloudTrail log files. Note, however, that CloudTrail does not record root sign-in failures.

The following record shows that an IAM user named Alice successfully signed into the AWS console without using multi-factor authentication.

```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAELOPP77CWZEXAMPLE",
        "arn": "arn:aws:iam::12345679012:user/alice",
        "accountId": "12345679012",
        "userName": "alice"
      },
      "eventTime": "2014-07-08T17:35:32Z",
      "eventSource": "signin.amazonaws.com",
      "eventName": "ConsoleLogin",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4b) Gecko/20030516 Mozilla Firebird/0.6",
      "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"
    }
  ]
}
```

The following record shows that an IAM user named Alice logged into the AWS console by using multi-factor authentication.

```
{
  "Records": [
    {
```

```
    "eventVersion": "1.02",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIDAEZ7VBM6PDZEXAMPLE",
      "arn": "arn:aws:iam::12345679012:user/Alice",
      "accountId": "12345679012",
      "userName": "Alice"
    },
    "eventTime": "2014-07-08T17:36:03Z",
    "eventSource": "signin.amazonaws.com",
    "eventName": "ConsoleLogin",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4b)
Gecko/20030516 Mozilla Firebird/0.6",
    "requestParameters": null,
    "responseElements": {
      "ConsoleLogin": "Success"
    },
    "additionalEventData": {
      "MobileVersion": "Yes",
      "LoginTo": "https://console.aws.amazon.com/sns",
      "MFAUsed": "Yes"
    },
    "eventID": "5d2c2f55-3d1e-4336-b940-dbf8e66f588c",
    "eventType": "AwsConsoleSignin"
  }
]
}
```

The following record shows an unsuccessful AWS console sign-in attempt because of an authentication failure.

```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAELOPP77CWZEXAMPLE",
        "accountId": "12345679012",
        "accessKeyId": "",
        "userName": "alice"
      },
      "eventTime": "2014-07-08T17:35:27Z",
      "eventSource": "signin.amazonaws.com",
      "eventName": "ConsoleLogin",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4b)
Gecko/20030516 Mozilla Firebird/0.6",
      "errorMessage": "Failed authentication",
      "requestParameters": null,
      "responseElements": {
        "ConsoleLogin": "Failure"
      },
    },
  ],
}
```

```
    "additionalEventData": {  
      "MobileVersion": "No",  
      "LoginTo": "https://console.aws.amazon.com/sns",  
      "MFAUsed": "No"  
    },  
    "eventID": "11ea990b-4678-4bcd-8fbe-62509088b7cf",  
    "eventType": "AwsConsoleSignin"  
  }  
]  
}
```

# CloudTrail Document History

The following table describes the documentation release history of AWS CloudTrail.

- **API version:** 2013-11-01
- **Latest documentation update:** September 1, 2015

Change	Description	Release Date
Added service support	This release supports Amazon S3 bucket level events. See <a href="#">Storage and Content Delivery (p. 11)</a> .	September 1, 2015
Added service support	This release supports AWS Device Farm. See <a href="#">Mobile Services (p. 9)</a> .	July 13, 2015
Added service support	This release supports Amazon API Gateway. See <a href="#">Application Services (p. 6)</a> .	July 9, 2015
Added service support	This release supports AWS CodePipeline. See <a href="#">Developer Tools (p. 8)</a> .	July 9, 2015
Added service support	This release supports Amazon DynamoDB. See <a href="#">Data-base (p. 7)</a> .	May 28, 2015
Added service support	This release supports CloudWatch Logs in the US West (N. California) region. See the <a href="#">CloudTrail release notes</a> . For more information about CloudTrail support for CloudWatch Logs monitoring, see <a href="#">Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 92)</a> .	May 19, 2015
Added service support	This release supports AWS Directory Service. See <a href="#">Security and Identity (p. 10)</a> .	May 14, 2015
Added service support	This release supports Amazon Simple Email Service (Amazon SES). See <a href="#">Application Services (p. 6)</a> .	May 7, 2015
Added service support	This release supports Amazon EC2 Container Service (see <a href="#">Compute (p. 7)</a> ).	April 9, 2015
Added service support	This release supports AWS Lambda (see <a href="#">Compute (p. 7)</a> ).	April 9, 2015

Change	Description	Release Date
Added service support	This release supports Amazon WorkSpaces. See <a href="#">Enterprise Applications (p. 8)</a> .	April 9, 2015
Added functionality and documentation	This release supports the lookup of AWS 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 AWS Command Line Interface (AWS CLI), or the AWS SDK. For more information, see <a href="#">Viewing CloudTrail Events (p. 30)</a> .	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 EU (Frankfurt) regions. Additional CloudWatch alarm examples have been added to <a href="#">Creating CloudWatch Alarms for CloudTrail Events</a> , and a new page has been added: <a href="#">Using a AWS CloudFormation Template to Create CloudWatch Alarms</a> .	March 5, 2015
Added API support	This release supports Amazon EC2 Simple Systems Manager (SSM). SSM lets you configure, manage and easily deploy custom Windows instance configurations. For more information about SSM, see <a href="#">Managing Windows Instance Configuration</a> . For information about the SSM API calls logged by CloudTrail, see <a href="#">Logging SSM API Calls Using AWS CloudTrail</a> .	February 17, 2015
New documentation	A new section that describes CloudTrail support for AWS Security Token Service (AWS STS) regional endpoints has been added to the <a href="#">CloudTrail Concepts</a> page.	February 17, 2015
Added service support	This release supports Amazon Route 53. See <a href="#">Networking (p. 10)</a> .	February 11, 2015
Added service support	This release supports AWS Config. See <a href="#">Management Tools (p. 9)</a> .	February 10, 2015
Added service support	This release supports AWS CloudHSM. See <a href="#">Security and Identity (p. 10)</a> .	January 8, 2015
Added service support	This release supports AWS CodeDeploy. See <a href="#">Developer Tools (p. 8)</a> .	December 17, 2014
Added service support	This release supports AWS Storage Gateway. See <a href="#">Storage and Content Delivery (p. 11)</a> .	December 16, 2014
Added region support	This release supports one additional region: us-gov-west-1 (AWS GovCloud (US)). See <a href="#">CloudTrail Supported Regions (p. 13)</a> .	December 16, 2014
Added service support	This release supports Amazon Glacier. See <a href="#">Storage and Content Delivery (p. 11)</a> .	December 11, 2014
Added service support	This release supports AWS Data Pipeline. See <a href="#">Analytics (p. 5)</a> .	December 2, 2014
Added service support	This release supports AWS Key Management Service. See <a href="#">Security and Identity (p. 10)</a> .	November 12, 2014

Change	Description	Release Date
New documentation	A new section, <a href="#">Monitoring CloudTrail Log Files with Amazon CloudWatch Logs (p. 92)</a> , 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, <a href="#">Using the CloudTrail Processing Library (p. 135)</a> , has been added to the guide. It provides information about how to write a CloudTrail log processor in Java using the AWS CloudTrail Processing Library.	November 5, 2014
Added service support	This release supports Amazon Elastic Transcoder. See <a href="#">Application Services (p. 6)</a> .	October 27, 2014
Added region support	This release supports one additional region: eu-central-1 (EU (Frankfurt)). See <a href="#">CloudTrail Supported Regions (p. 13)</a> .	October 23, 2014
Added service support	This release supports Amazon CloudSearch. See <a href="#">Application Services (p. 6)</a> .	October 16, 2014
Added service support	This release supports Amazon Simple Notification Service. See <a href="#">Mobile Services (p. 9)</a> .	October 09, 2014
Added service support	This release supports Amazon ElastiCache. See <a href="#">Database (p. 7)</a> .	September 15, 2014
Added service support	This release supports Amazon WorkDocs. See <a href="#">Enterprise Applications (p. 8)</a> .	August 27, 2014
Added new content	This release includes a topic that discusses logging sign-in events. See <a href="#">AWS Console Sign-in Events (p. 151)</a> .	July 24, 2014
Added new content	The <b>eventVersion</b> element for this release has been upgraded to version 1.02 and three new fields have been added. See <a href="#">CloudTrail Record Contents (p. 144)</a> .	July 18, 2014
Added service support	This release supports Auto Scaling (see <a href="#">Compute (p. 7)</a> ) and Amazon SQS (see <a href="#">Application Services (p. 6)</a> ).	July 17, 2014
Added region support	This release supports three additional regions: ap-south-east-1 (Asia Pacific (Singapore)), ap-northeast-1 (Asia Pacific (Tokyo)), sa-east-1 (South America (Sao Paulo)). See <a href="#">CloudTrail Supported Regions (p. 13)</a> .	June 30, 2014
Additional service support	This release supports Amazon Redshift. See <a href="#">Analytics (p. 5)</a> .	June 10, 2014
Added service support	This release supports AWS OpsWorks. See <a href="#">Management Tools (p. 9)</a> .	June 5, 2014
Added service support	This release supports Amazon CloudFront. See <a href="#">Storage and Content Delivery (p. 11)</a> .	May 28, 2014
Added region support	This release supports three additional regions: us-west-1 (US West (N. California)), eu-west-1 (EU (Ireland)), ap-southeast-2 (Asia Pacific (Sydney)). See <a href="#">CloudTrail Supported Regions (p. 13)</a> .	May 13, 2014



Change	Description	Release Date
Added service support	This release supports Amazon Simple Workflow Service. See <a href="#">Application Services (p. 6)</a> .	May 9, 2014
Added new content	This release includes topics that discuss sharing log files between accounts. See <a href="#">Sharing CloudTrail Log Files Between AWS Accounts (p. 81)</a> .	May 2, 2014
Added service support	This release supports Amazon CloudWatch. See <a href="#">Management Tools (p. 9)</a> .	April 28, 2014
Added service support	This release supports Amazon Kinesis. See <a href="#">Analytics (p. 5)</a> .	April 22, 2014
Added service support	This release supports AWS Direct Connect. See <a href="#">Networking (p. 10)</a> .	April 11, 2014
Added service support	This release supports Amazon Elastic MapReduce. See <a href="#">Analytics (p. 5)</a> .	April 4, 2014
Added service support	This release supports Elastic Beanstalk. See <a href="#">Compute (p. 7)</a> .	April 2, 2014
Additional service support	This release supports AWS CloudFormation. See <a href="#">Management Tools (p. 9)</a> .	March 7, 2014
New guide	This release introduces AWS CloudTrail.	November 13, 2013