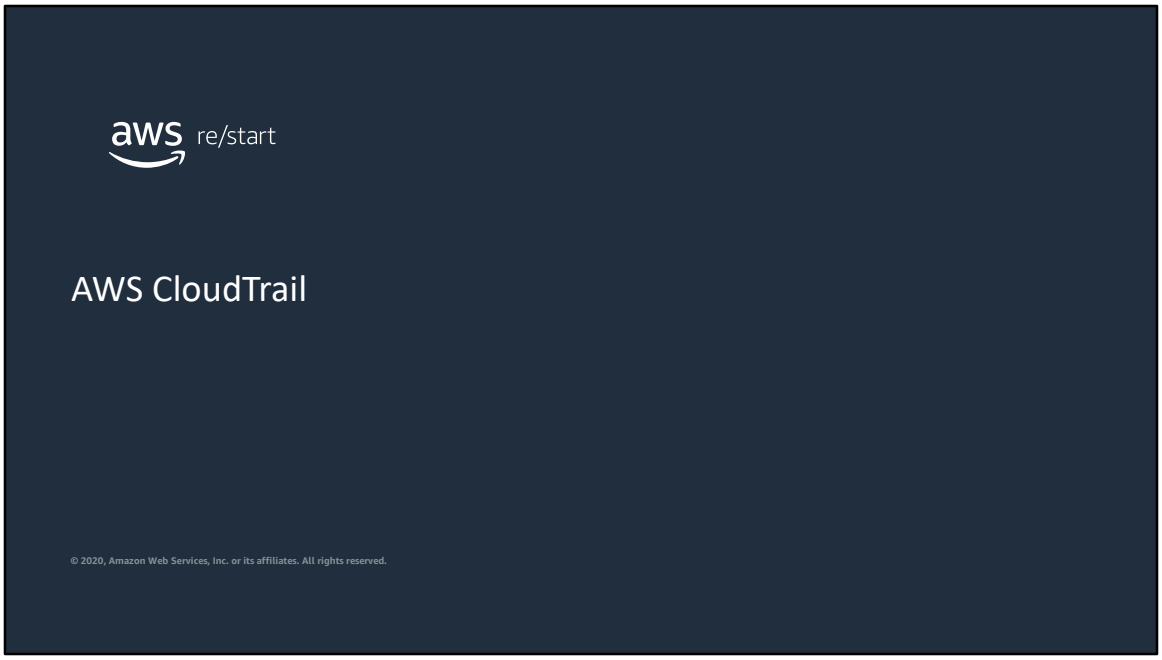


The lesson will now explore AWS CloudTrail.

What you will learn

At the core of the lesson

You will learn how to:

- Explain the purpose and function of AWS CloudTrail

Topics:

- AWS CloudTrail

Key terms:

- Log entry
- requestParameters
- responseElements

aws re/start

At the end of this module, you will be able to:

- Explain the purpose and function of AWS CloudTrail.

AWS CloudTrail

AWS CloudTrail is a service that:

- Logs, continuously monitors, and retains account activity that is related to actions across your AWS infrastructure
- Records application programming interface (API) calls for most AWS services
  - AWS Management Console and AWS Command Line Interface (AWS CLI) activity are also recorded
- Is supported for a growing number of AWS services
- Automatically pushes logs to Amazon Simple Storage Service (Amazon S3) after it is configured
- Will not track events within an Amazon Elastic Compute Cloud (Amazon EC2) instance
  - Example: Manual shutdown of an instance

AWS CloudTrail

aws re/start

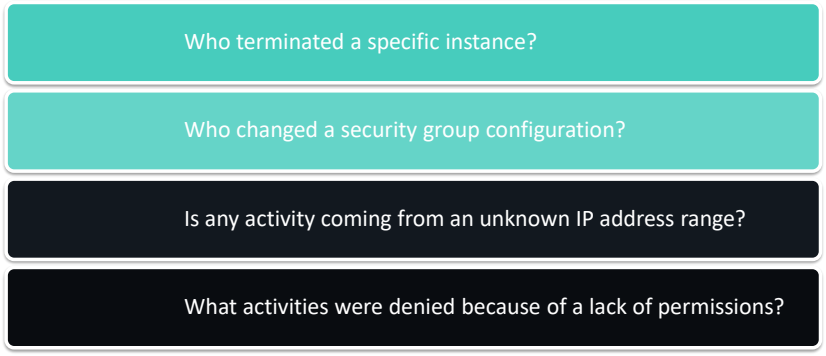
AWS CloudTrail is an AWS service that generates logs of calls to the AWS application programming interface (API). The AWS API underlies both the AWS Command Line Interface (AWS CLI) and the AWS Management Console. Thus, CloudTrail can record all activity against the services that it monitors. It enables governance, compliance, operational auditing, and risk auditing of AWS accounts.

A large (and growing) number of AWS services are supported. For details, refer to CloudTrail Supported Services and Integrations.

After CloudTrail is configured, it pushes the auditing logs to Amazon Simple Storage Service (Amazon S3). Though AWS CloudTrail is full-featured, it does not track events that occur *within* an Amazon Elastic Compute Cloud (Amazon EC2) instance. For example, CloudTrail does not track when someone manually shuts down an instance by using a Secure Shell (SSH) session connection to the instance. They could issue a command such as `sudo shutdown -h now`.

CloudTrail example

CloudTrail can help you answer questions that require detailed analysis.



aws re/start

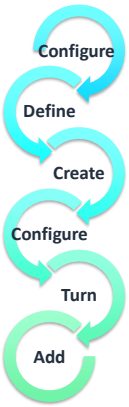
By using CloudTrail, you can store logs on API usage in an S3 bucket. Later, you can analyze those logs to answer a number of questions, such as:

- Why was a long-running instance terminated, and who terminated it? The answers could be useful for organizational traceability and accountability.
- Who changed a security group configuration? Accountability and security auditing teams might want to know this information.
- Is any activity coming from an unknown IP address range? Such activity could indicate a potential external attack, which is a security concern.
- What activities were denied because of a lack of permissions? Such activity could indicate internal or external attack attempts.

Configure a trail

The steps to configure a trail are:

1. Configure a new or existing Amazon Simple Storage Service (Amazon S3) bucket for uploading log files.
2. Define a trail to log desired events (all management events are logged by default).
3. Create an Amazon Simple Notification Service (Amazon SNS) topic to receive notifications.
4. Configure Amazon CloudWatch Logs to receive logs from CloudTrail (optional).
5. Turn on log file encryption and integrity validation for log files (optional).
6. Add tags to your trail (optional).



5

aws re/start

By default, when you access the CloudTrail event history for the Region that you are viewing, CloudTrail shows only the results from the last 90 days. These events are limited to management events with create, modify, and delete API calls; and also account activity. For a *complete record* of account activity—including all management events, data events, and read-only activity—you must configure a CloudTrail *trail*.

You can create a trail by using the CloudTrail console or by using the AWS CLI.

The following options can be configured:

1. Create an S3 bucket or specify an existing bucket where you want the log files to be stored.
2. Configure your trail to log *read-only*, *write-only*, or *all* management and data events. By default, trails log all management events.
3. Create an Amazon Simple Notification Service (Amazon SNS) topic to receive notifications when log files are delivered.
4. Optionally, configure Amazon CloudWatch Logs to receive logs from CloudTrail so that you can monitor for specific log events.
5. Optionally, turn on *log file encryption* for added security.
6. Optionally, add tags (custom key-value pairs) to your trail.

For more information, refer to What Is AWS CloudTrail?

CloudTrail log entry: requestParameters

This example shows a **requestParameters** section of a log entry. It includes parameters, such as:

- **userIdentity** – Who (or what application) took an action
- **eventTime** – The date and time that the action occurred
- **eventSource** – An indication of whether the action was done through the console, the AWS CLI, or an API call

```
{   "eventVersion" : "1.01",
    "userIdentity" : {
        "type" : "IAMUser",
        "principalId" : "AIDAyyyyyyyyyyyyyyyy",
        "arn" : "arn:aws:iam::xxxxxxxxxxxx:user/tests3user",
        "accountId" : "xxxxxxxxxxxx",
        "userName" : "tests3user"
    },
    "eventTime" : "2018-09-23T22:41:38Z",
    "eventSource" : "signin.amazonaws.com",
    "eventName" : "ConsoleLogin",
    "awsRegion" : "us-east-1",
    "sourceIPAddress" : "54.240.217.10",
    "userAgent" : "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0)
Gecko/20100101 Firefox/31.0",
```


aws re/start

Each JavaScript Object Notation (JSON)-formatted CloudTrail log file can contain one or more log entries.

A *log entry* represents a single request from any source. It includes information about the requested action and response.

The example shows the **requestParameters** section of a log entry. It includes parameters, such as:

- **userIdentity** – Who (or what application) took an action.
- **eventTime** – The date and time the action occurred.
- **eventSource** – An indication of whether the action was done through the console, the AWS CLI, or an API call.

Additional parameters can also be included. The actual list of parameters depends on the type of action that was logged.

Log entries are not guaranteed to be in any particular order. They are not an ordered stack trace of API calls.

CloudTrail log entry: responseElements (success example)

This example shows the **responseElements** section of the same sample log entry.


```
"responseElements" : {
                "ConsoleLogin" : "Success"
            },
            "additionalEventData" : {
                "MobileVersion" : "No",
                "LoginTo" :
"https://console.aws.amazon.com/console/home?state\u003dhas
hArgs%23\u0026isauthcode\u003dtrue",
                "MFAUsed" : "No"
            },
            "eventID" : "b31716e9-13e5-4fb5-9c60-
1c723c59f5a6"
        }
```


aws re/start

This example shows the **responseElements** section of the same sample log entry.

In this case, the attempt to log in to the console was successful. The user did not use the mobile version of the console, and the user also used multi-factor authentication (MFA) to log in.

CloudTrail log entry: responseElements (failure example)

This example is the **responseElements** section of a log in attempt for a different sample log entry.


```
        "errorMessage" : "Failed authentication",
            "requestParameters" : null,
            "responseElements" : {
                "ConsoleLogin" : "Failure"
            },
            "additionalEventData" : {
                "MobileVersion" : "No",
                "LoginTo" :
"https://console.aws.amazon.com/console/home?state
\u003dhashArgs%23\u0026isauthcode\u003dtrue",
                "MFAUsed" : "No"
            },
            "eventID" : "99dc0ee0-ec1d-4a14-bb74-
10407b3b8ab1"
        },
```


aws re/start

This example is the **responseElements** section of a login attempt for a different sample log entry.

In this example, the login attempt was unsuccessful.

Monitoring and security

Examine CloudWatch Logs and CloudTrail to detect potential unauthorized use.

Examples:

- Failed AWS Management Console sign-in attempts, or sign-in attempts from suspicious IP addresses
- Unauthorized access to services that use the API
- Suspicious launches of resources

When you monitor the activity on your account and secure your resources and data, the features of CloudWatch and CloudTrail are complementary. Using both services is a best practice. For example, you can examine the logs from CloudWatch Logs and also examine CloudTrail entries to detect potential unauthorized use.

Other example uses of these services include:

- Monitoring for failed AWS Management Console sign-in attempts, especially sign-in attempts from suspicious IP addresses
- Detecting unauthorized access to services through API calls
- Identifying a suspicious launching of AWS resources

Key takeaways

- AWS CloudTrail enables the auditing of AWS accounts by continuously logging account activity.
- The information that CloudTrail captures can be sent to storage services like Amazon S3, or to business reporting tools.
- Monitoring logs is an integral part of a professional security posture.

aws re/start

The key takeaway from this section is:

- AWS CloudTrail enables auditing of AWS accounts by continuously logging account activity.
- The information that CloudTrail captures can be sent to storage services like Amazon S3, or to business reporting tools.
- Monitoring logs is an integral part of a professional security posture.