Logging IAM and AWS STS API calls with AWS CloudTrail

Archived: 2026-04-05 13:40:38 UTC

IAM and AWS STS are integrated with AWS CloudTrail, a service that provides a record of actions taken by an IAM user or role. CloudTrail captures all API calls for IAM and AWS STS as events, including calls from the console and from API calls. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. You can use CloudTrail to get information about the request that was made to IAM or AWS STS. For example, you can view the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Topics

IAM and AWS STS information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in IAM or AWS STS, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for IAM and AWS STS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers

the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All IAM and AWS STS actions are logged by CloudTrail and are documented in the IAM API Reference and the AWS Security Token Service API Reference.

Logging IAM and AWS STS API requests

CloudTrail logs all authenticated API requests to IAM and AWS STS API operations. CloudTrail also logs non-authenticated requests to the AWS STS actions, `AssumeRoleWithSAML` and `AssumeRoleWithWebIdentity`, and logs information provided by the identity provider. However, some non-authenticated AWS STS requests might not be logged because they do not meet the minimum expectation of being sufficiently valid to be trusted as a legitimate request. For cross-account role assumption requests, CloudTrail does not log denied AWS STS requests in the target account's CloudTrail.

You can use the logged information to map calls made by an OIDC or SAML federated principal with an assumed role back to the originating external federated caller. In the case of `AssumeRole`, you can map calls back to the originating AWS service or to the account of the originating user. The `userIdentity` section of the JSON data in the CloudTrail log entry contains the information that you need to map the AssumeRole* request with a specific session principal. For more information, see CloudTrail userIdentity Element in the *AWS CloudTrail User Guide*.

AWS CloudTrail logs will contain MFA information when the IAM user sign in with MFA. If the IAM user assumes an IAM role, CloudTrail will also log `mfaAuthenticated: true` in the `sessionContext` attributes for actions performed using the assumed role. However, CloudTrail logging is separate from what IAM requires when API calls are made with the assumed role's credentials. For more information, see CloudTrail userIdentity Element.

For example, calls to the IAM `CreateUser`, `DeleteRole`, `ListGroups`, and other API operations are all logged by CloudTrail.

Examples for this type of log entry are presented later in this topic.

Logging API requests to other AWS services

Authenticated requests to other AWS service API operations are logged by CloudTrail, and these log entries contain information about who generated the request.

For example, assume that you made a request to list Amazon EC2 instances or create an AWS CodeDeploy deployment group. Details about the person or service that made the request are contained in the log entry for that request. This information helps you determine whether the request was made by the AWS account root user, an IAM user, a role, or another AWS service.

For more details about the user identity information in CloudTrail log entries, see userIdentity Element in the *AWS CloudTrail User Guide*.

Logging user sign-in events

CloudTrail logs sign-in events to the AWS Management Console, local development tools like AWS CLI and SDKs, the AWS discussion forums, and AWS Marketplace. CloudTrail logs successful and failed sign-in attempts for IAM users, SAML and OIDC federated principals, and AWS STS federated user principals.

To view sample CloudTrail events for successful and unsuccessful root user sign-ins, see Example event records for root users in the *AWS CloudTrail User Guide*.

As a security best practice, AWS does not log the entered IAM user name text when the sign-in failure is caused by *an incorrect user name*. The user name text is masked by the value `HIDDEN_DUE_TO_SECURITY_REASONS`. For an example of this, see Example sign-in failure event caused by incorrect user name, later in this topic. The user name text is obscured because such failures might be caused by user errors. Logging these errors could expose potentially sensitive information. For example:

- You accidentally type your password in the user name box.
- You choose the link for the sign-in page of one AWS account, but then type the account number for a different AWS account.
- You forget which account you are signing in to and accidentally type the account name of your personal email account, your bank sign-in identifier, or some other private ID.

Logging sign-in events for temporary credentials

When a principal requests temporary credentials, the principal type determines how CloudTrail logs the event. This can be complicated when a principal assumes a role in another account. There are multiple API calls to perform operations related to role cross-account operations. First, the principal calls an AWS STS API to retrieve the temporary credentials. That operation is logged in the calling account and the account where the AWS STS operation is performed. Then the principal then uses the role to perform other API calls in the assumed role's account.

You can use the `sts:SourceIdentity` condition key in the role trust policy to require users to specify an identity when they assume a role. For example, you can require that IAM users specify their own user name as their source identity. This can help you determine which user performed a specific action in AWS. For more information, see sts:SourceIdentity. You can also use sts:RoleSessionName to require users to specify a session name when they assume a role. This can help you differentiate between role sessions for a role that is used by different principals when you review AWS CloudTrail logs.

The following table shows how CloudTrail logs different user identity information for each of the AWS STS APIs that generate temporary credentials.

Principal type	**STS API**	**User identity in CloudTrail log for caller's account**	**User identity in CloudTrail log for the assumed role's account**	**User identity in CloudTrail log for the role's subsequent API calls**
AWS account root user credentials	GetSessionToken	Root user identity	Role owner account is same as calling account	Root user identity
AWS account root user credentials	AssumeRoot	Root user session	Account number and principal ID (if a user)	Root user session
IAM user	GetSessionToken	IAM user identity	Role owner account is same as calling account	IAM user identity
IAM user	GetFederationToken	IAM user identity	Role owner account is same as calling account	IAM user identity
IAM user	AssumeRole	IAM user identity	Account number and principal ID (if a user), or AWS service principal	Role identity only (no user)
Externally authenticated user	AssumeRoleWithSAML	n/a	SAML user identity	Role identity only (no user)
Externally authenticated user	AssumeRoleWithWebIdentity	n/a	OIDC/Web user identity	Role identity only (no user)

CloudTrail considers an action read-only if it does not have any mutating effect on a resource. When logging a read-only event, CloudTrail redacts the `responseElements` information in the log. When CloudTrail logs an event that is not read-only, the full `responseElements` is shown in the log entry. For the AWS STS APIs

`AssumeRole`, `AssumeRoleWithSAML`, and `AssumeRoleWithWebIdentity`, even though they are logged as read-only, CloudTrail will include the full `responseElements` except `secretAccessKey` in the log for these APIs.

The following table shows how CloudTrail logs `responseElements` and `readOnly` information for each of the AWS STS APIs that generate temporary credentials.

STS API	Response elements information	Read-only
AssumeRole	Included	true
AssumeRoleWithSAML	Included	true
AssumeRoleWithWebIdentity	Included	true
AssumeRoot	Included	false
GetFederationToken	Included	false
GetSessionToken	Included	false

Example IAM API events in CloudTrail log

CloudTrail log files contain events that are formatted using JSON. An API event represents a single API request and includes information about the principal, the requested action, any parameters, and the date and time of the action.

Example IAM API event in CloudTrail log file

The following example shows a CloudTrail log entry for a request made for the IAM `GetUserPolicy` action.

```
{
   "eventVersion": "1.09",
   "userIdentity": {
       "type": "AssumedRole",
       "principalId": "AIDACKCEVSQ6C2EXAMPLE:Role-Session-Name",
       "arn": "arn:aws:sts::111122223333:assumed-role/Role-Name/Role-Session-Name",
       "accountId": "111122223333",
       "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
       "sessionContext": {
           "sessionIssuer": {
               "type": "Role",
               "principalId": "AIDACKCEVSQ6C2EXAMPLE",
               "arn": "arn:aws:iam::111122223333:role/Admin",
               "accountId": "111122223333",
               "userName": "Admin"
           },
           "attributes": {
```

```
                "creationDate": "2024-09-09T17:50:16Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-09-09T17:51:44Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "GetUserPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.101",
    "userAgent": "aws-cli/1.16.96 Python/2.7.8 Linux/10 botocore/1.12.86",
    "requestParameters": {
        "userName": "ExampleIAMUserName",
        "policyName": "ExamplePoliccyName"
    },
    "responseElements": null,
    "requestID": "9EXAMPLE-0c68-11e4-a24e-d5e16EXAMPLE",
    "eventID": "cEXAMPLE-127e-4632-980d-505a4EXAMPLE",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "iam.amazonaws.com"
    }
}
```

From this event information, you can determine that the request was made to get a user policy named `ReadOnlyAccess-JaneDoe-201407151307` for user `JaneDoe`, as specified in the `requestParameters` element. You can also see that the request was made by an IAM user named `JaneDoe` on July 15, 2014 at 9:40 PM (UTC). In this case, the request originated in the AWS Management Console, as you can tell from the `userAgent` element.

Example AWS STS API events in CloudTrail log

CloudTrail log files contain events that are formatted using JSON. An API event represents a single API request and includes information about the principal, the requested action, any parameters, and the date and time of the action.

Example cross-account AWS STS API events in CloudTrail log files

The IAM user named `John` in account 777788889999 calls the AWS STS AssumeRole action to assume the role `EC2-dev` in account 111122223333. The account administrator requires users to set a source identity equal to

their user name when assuming the role. The user passes in the source identity value of `John`.


```
{
 "eventVersion": "1.05",
 "userIdentity": {
   "type": "IAMUser",
   "principalId": "AIDAQRSTUVWXYZEXAMPLE",
   "arn": "arn:aws:iam::777788889999:user/John",
   "accountId": "777788889999",
   "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
   "userName": "John"
 },
 "eventTime": "2014-07-18T15:07:39Z",
 "eventSource": "sts.amazonaws.com",
 "eventName": "AssumeRole",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "192.0.2.101",
 "userAgent": "aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
 "requestParameters": {
   "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
   "roleSessionName": "John-EC2-dev",
   "sourceIdentity": "John",
   "serialNumber": "arn:aws:iam::777788889999:mfa"
 },
 "responseElements": {
   "credentials": {
     "sessionToken": " <encoded session token blob> ",
     "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
     "expiration": "Jul 18, 2023, 4:07:39 PM"
     },
   "assumedRoleUser": {
     "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:John-EC2-dev",
     "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/John-EC2-dev"
   },
 "sourceIdentity": "John"
 },
 "resources": [
   {
     "ARN": "arn:aws:iam::111122223333:role/EC2-dev",
     "accountId": "111122223333",
     "type": "AWS::IAM::Role"
   }
 ],
 "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
 "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
 "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE",
 "eventType": "AwsApiCall",
```

```
  "recipientAccountId": "111122223333"
}
```

The second example shows the assumed role account's (111122223333) CloudTrail log entry for the same request.

```
 {
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AWSAccount",
    "principalId": "AIDAQRSTUVWXYZEXAMPLE",
    "accountId": "777788889999"
  },
  "eventTime": "2014-07-18T15:07:39Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.101",
  "userAgent": "aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67",
  "requestParameters": {
    "roleArn": "arn:aws:iam::111122223333:role/EC2-dev",
    "roleSessionName": "John-EC2-dev",
    "sourceIdentity": "John",
    "serialNumber": "arn:aws:iam::777788889999:mfa"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": " <encoded session token blob> ",
      "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
      "expiration": "Jul 18, 2014, 4:07:39 PM"
    },
    "assumedRoleUser": {
      "assumedRoleId": "AIDAQRSTUVWXYZEXAMPLE:John-EC2-dev",
      "arn": "arn:aws:sts::111122223333:assumed-role/EC2-dev/John-EC2-dev"
      },
  "sourceIdentity": "John"
  },
  "requestID": "4EXAMPLE-0e8d-11e4-96e4-e55c0EXAMPLE",
  "sharedEventID": "bEXAMPLE-efea-4a70-b951-19a88EXAMPLE",
  "eventID": "dEXAMPLE-ac7f-466c-a608-4ac8dEXAMPLE"
}
```

Example AWS STS role chaining API event in CloudTrail log file

The following example shows a CloudTrail log entry for a request made by John Doe in account 111111111111. John previously used his `John` user to assume the `JohnRole1` role. For this request, he uses the credentials from that role to assume the `JohnRole2` role. This is known as role chaining. The source identity that he set when he

assumed the `John1` role persists in the request to assume `JohnRole2`. If John tries to set a different source identity when assuming the role, the request is denied. John passes two session tags into the request. He sets those two tags as transitive. The request inherits the `Department` tag as transitive because John set it as transitive when he assumed `JohnRole1`. For more information about source identity, see Monitor and control actions taken with assumed roles. For more information about transitive keys in role chains, see Chaining roles with session tags.


```
{
   "eventVersion": "1.05",
   "userIdentity": {
       "type": "AssumedRole",
       "principalId": "AROAIN5ATK5U7KEXAMPLE:JohnRole1",
       "arn": "arn:aws:sts::111111111111:assumed-role/John/JohnRole1",
       "accountId": "111111111111",
       "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
       "sessionContext": {
           "attributes": {
               "mfaAuthenticated": "false",
               "creationDate": "2019-10-02T21:50:54Z"
           },
               "sessionIssuer": {
               "type": "Role",
              "principalId": "AROAIN5ATK5U7KEXAMPLE",
               "arn": "arn:aws:iam::111111111111:role/JohnRole1",
               "accountId": "111111111111",
               "userName": "John"
           },
           "sourceIdentity": "John"
       }
   },
   "eventTime": "2019-10-02T22:12:29Z",
   "eventSource": "sts.amazonaws.com",
   "eventName": "AssumeRole",
   "awsRegion": "us-east-2",
   "sourceIPAddress": "123.145.67.89",
   "userAgent": "aws-cli/1.16.248 Python/3.4.7 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 botocore/1.12.239'
   "requestParameters": {
       "incomingTransitiveTags": {
           "Department": "Engineering"
       },
       "tags": [
           {
               "value": "johndoe@example.com",
               "key": "Email"
           },
           {
               "value": "12345",
```

```
                "key": "CostCenter"
            }
        ],
        "roleArn": "arn:aws:iam::111111111111:role/JohnRole2",
        "roleSessionName": "Role2WithTags",
        "sourceIdentity": "John",
        "transitiveTagKeys": [
            "Email",
            "CostCenter"
        ],
        "durationSeconds": 3600
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
            "expiration": "Oct 2, 2019, 11:12:29 PM",
            "sessionToken": "AgoJb3JpZ2luX2VjEB4aCXVzLXdlc3QtMSJHMEXAMPLETOKEN+//rJb8Lo30mFc5MlhFCEbubZvEj0wHB/r
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAIFR7WHDTSOYQYHFUE:Role2WithTags",
            "arn": "arn:aws:sts::111111111111:assumed-role/test-role/Role2WithTags"
        },
    "sourceIdentity": "John"
    },
    "requestID": "b96b0e4e-e561-11e9-8b3f-7b396EXAMPLE",
    "eventID": "1917948f-3042-46ec-98e2-62865EXAMPLE",
    "resources": [
        {
            "ARN": "arn:aws:iam::111111111111:role/JohnRole2",
            "accountId": "111111111111",
            "type": "AWS::IAM::Role"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```


Example AWS service AWS STS API event in CloudTrail log file

The following example shows a CloudTrail log entry for a request made by an AWS service calling another service API using permissions from a service role. It shows the CloudTrail log entry for the request made in account 777788889999.


```
 {
  "eventVersion": "1.04",
  "userIdentity": {
```

```
    "type": "AssumedRole",
    "principalId": "AROAQRSTUVWXYZEXAMPLE:devdsk",
    "arn": "arn:aws:sts::777788889999:assumed-role/AssumeNothing/devdsk",
    "accountId": "777788889999",
    "accessKeyId": "ASIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-11-14T17:25:26Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAQRSTUVWXYZEXAMPLE",
        "arn": "arn:aws:iam::777788889999:role/AssumeNothing",
        "accountId": "777788889999",
        "userName": "AssumeNothing"
      }
    }
  },
  "eventTime": "2016-11-14T17:25:45Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "DeleteBucket",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.1",
  "userAgent": "[aws-cli/1.11.10 Python/2.7.8 Linux/3.2.45-0.6.wd.865.49.315.metal1.x86_64 botocore/1.4.67]",
  "requestParameters": {
    "bucketName": "amzn-s3-demo-bucket"
  },
  "responseElements": null,
  "requestID": "EXAMPLE463D56D4C",
  "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "777788889999"
}
```


Example SAML AWS STS API event in CloudTrail log file

The following example shows a CloudTrail log entry for a request made for the AWS STS AssumeRoleWithSAML action. The request includes the SAML attributes `CostCenter` and `Project` that are passed through the SAML assertion as session tags. Those tags are set as transitive so that they persist in role chaining scenarios. The request includes the optional API parameter `DurationSeconds`, represented as `durationSeconds` in the CloudTrail log, and is set to `1800` seconds. The request also includes the SAML attribute `sourceIdentity`, which is passed in the SAML assertion. If someone uses the resulting role session credentials to assume another role, this source identity persists.

```
{
   "eventVersion": "1.08",
   "userIdentity": {
       "type": "SAMLUser",
       "principalId": "SampleUkh1i4+ExamplexL/jEvs=:SamlExample",
       "userName": "SamlExample",
       "identityProvider": "bdGOnTesti4+ExamplexL/jEvs="
   },
   "eventTime": "2023-08-28T18:30:58Z",
   "eventSource": "sts.amazonaws.com",
   "eventName": "AssumeRoleWithSAML",
   "awsRegion": "us-east-2",
   "sourceIPAddress": "AWS Internal",
   "userAgent": "aws-internal/3 aws-sdk-java/1.12.479 Linux/5.10.186-157.751.amzn2int.x86_64 OpenJDK_64-Bit_Se
   "requestParameters": {
       "sAMLAssertionID": "_c0046cEXAMPLEb9d4b8eEXAMPLE2619aEXAMPLE",
       "roleSessionName": "MyAssignedRoleSessionName",
       "sourceIdentity": "MySAMLUser",
       "principalTags": {
           "CostCenter": "987654",
           "Project": "Unicorn",
           "Department": "Engineering"
       },
       "transitiveTagKeys": [
           "CostCenter",
           "Project"
       ],
       "roleArn": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth",
       "principalArn": "arn:aws:iam::444455556666:saml-provider/Shibboleth",
       "durationSeconds": 1800
   },
   "responseElements": {
       "credentials": {
           "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
           "sessionToken": " <encoded session token blob> ",
           "expiration": "Aug 28, 2023, 7:00:58 PM"
       },
       "assumedRoleUser": {
           "assumedRoleId": "AROAD35QRSTUVWEXAMPLE:MyAssignedRoleSessionName",
           "arn": "arn:aws:sts::444455556666:assumed-role/SAMLTestRoleShibboleth/MyAssignedRoleSessionName"
       },
       "packedPolicySize": 1,
       "subject": "SamlExample",
       "subjectType": "transient",
       "issuer": "https://server.example.com/idp/shibboleth",
       "audience": "https://signin.aws.amazon.com/saml",
```

```
        "nameQualifier": "bdGOnTesti4+ExamplexL/jEvs=",
        "sourceIdentity": "MySAMLUser"
    },
    "requestID": "6EXAMPLE-e595-11e5-b2c7-c974fEXAMPLE",
    "eventID": "dEXAMPLE-265a-41e0-9352-4401bEXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::444455556666:role/SAMLTestRoleShibboleth"
        },
        {
            "accountId": "444455556666",
            "type": "AWS::IAM::SAMLProvider",
            "ARN": "arn:aws:iam::444455556666:saml-provider/test-saml-provider"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "444455556666",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "sts.us-east-2.amazonaws.com"
    }
}
```

Example OIDC AWS STS API event in CloudTrail log file

The following example shows a CloudTrail log entry for a request made for the AWS STS AssumeRoleWithWebIdentity action. The request includes the attributes `CostCenter` and `Project` that are passed through the OpenID Connect (OIDC) identity provider (IdP) token as session tags. Those tags are set as transitive so that they persist in role chaining. The request includes the `sourceIdentity` attribute from the identity provider token. If someone uses the resulting role session credentials to assume another role, this source identity persists.

The CloudTrail log entry also contains an `additionalEventData` field with an `identityProviderConnectionVerificationMethod` attribute. This attribute indicates the method AWS used to verify the connection with the OIDC provider. The attribute value will be either `IAMTrustStore` or `Thumbprint`. The `IAMTrustStore` value indicates that AWS successfully verified the connection with the OIDC IdP using our library of trusted root certificate authorities (CAs). The `Thumbprint` value indicates that AWS used a certificate thumbprint set in the IdP configuration to verify the OIDC IdP server certificate.

```
{
 "eventVersion": "1.08",
 "userIdentity": {
   "type": "WebIdentityUser",
   "principalId": "arn:aws:iam::444455556666:oidc-provider/ <issuer url of OIDC provider> : <id of application> : <
   "userName": " <id of user> ",
   "identityProvider": "arn:aws:iam::444455556666:oidc-provider/ <issuer url of OIDC provider> "
 },
 "eventTime": "2024-07-09T15:41:37Z",
 "eventSource": "sts.amazonaws.com",
 "eventName": "AssumeRoleWithWebIdentity",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "192.0.2.101",
 "userAgent": "aws-cli/2.13.29 Python/3.11.6 Windows/10 exe/AMD64 prompt/off command/sts.assume-role-with-web-i
 "requestParameters": {
   "roleArn": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole",
   "roleSessionName": " <assigned role session name> ",
   "sourceIdentity": "MyWebIdentityUser",
   "durationSeconds": 3600,
   "principalTags": {
     "CostCenter": "24680",
     "Project": "Pegasus"
   },
   "transitiveTagKeys": [
     "CostCenter",
     "Project"
   ]
 },
 "responseElements": {
   "credentials": {
     "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
     "sessionToken": " <encoded session token blob> ",
     "expiration": "Jul 9, 2024, 4:41:37 PM"
   },
   "subjectFromWebIdentityToken": " <id of user> ",
   "sourceIdentity": "MyWebIdentityUser",
   "assumedRoleUser": {
     "assumedRoleId": "AROA123456789EXAMPLE: <assigned role session name> ",
     "arn": "arn:aws:sts::444455556666:assumed-role/FederatedWebIdentityRole/ <assigned role session name> "
   },
   "provider": "arn:aws:iam::444455556666:oidc-provider/ <issuer url of OIDC provider> ",
   "audience": " <id of application> "
 },
 "additionalEventData": {
   "identityProviderConnectionVerificationMethod": "IAMTrustStore"
 },
```

```
  "requestID": "aEXAMPLE-0b26-40df-8973-c7012EXAMPLE",
  "eventID": "aEXAMPLE-ee29-4ac0-a0ed-3f5c5EXAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "444455556666",
      "type": "AWS::IAM::Role",
      "ARN": "arn:aws:iam::444455556666:role/FederatedWebIdentityRole"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "444455556666",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "sts.us-east-2.amazonaws.com"
  }
}
```

Example AWS STS API event using the global endpoint in CloudTrail log file

For requests to the AWS Security Token Service (AWS STS) global endpoint (`https://sts.amazonaws.com`), AWS STS includes additional AWS CloudTrail log fields: `endpointType` and `awsServingRegion` . These fields appear under the `addtionalEventData` `RequestDetails` element to log the serving AWS Region and endpoint type being called. The `endpointType` field can have a value of `global` or `regional` to indicate the type of global endpoint that served the request. For more information about the AWS STS global endpoint changes, see [AWS STS Regions and endpoints](#).

Note

AWS CloudTrail logs for requests made to the AWS STS global endpoint will be sent to the US East (N. Virginia) Region. CloudTrail logs for requests served by AWS STS Regional endpoints will continue to be logged to their respective Region in CloudTrail.

The following example shows a CloudTrail log entry for an AWS STS request made to the global endpoint (`https://sts.amazonaws.com`) that originated from the Europe (Stockholm) Region - eu-north-1. The `endpointType` field value of `global` indicates that the AWS STS request was served by the global endpoint in the Europe (Stockholm) Region.

```
{
   "eventVersion": "1.08",
   "userIdentity": {
       "type": "AssumedRole",
```

```
        "principalId": "AROA123456789EXAMPLE:developer",
        "arn": "arn:aws:sts::777788889999:assumed-role/Admin/developer",
        "accountId": "777788889999",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::777788889999:role/Admin",
                "accountId": "777788889999",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2025-02-12T21:44:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-02-12T22:16:48Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/2.15.33 Python/3.11.8 Linux/5.10.233-204.894.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/
    "requestParameters": {
        "roleArn": "arn:aws:iam::777788889999:role/test-role",
        "roleSessionName": "test-global-assume-role"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": " <encoded session token blob> ",
            "expiration": "Feb 12, 2025, 11:16:48 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROA987654321EXAMPLE:test-global-assume-role",
            "arn": "arn:aws:sts::777788889999:assumed-role/test-role/test-global-assume-role"
        }
    },
    "additionalEventData": {
        "RequestDetails": {
            "awsServingRegion": "eu-north-1",
            "endpointType": "global"
        }
    },
    "requestID": "EXAMPLE7-2497-457a-9586-f21feEXAMPLE",
```

```
    "eventID": "EXAMPLEc-3d26-4c3a-9c94-722a9EXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "777788889999",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::777788889999:role/test-role"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "777788889999",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "sts-global.eu-north-1.amazonaws.com"
    }
}
```

For comparison, the following example shows a CloudTrail log entry for an AWS STS request made to the Regional endpoint (`https://sts.us-west-2.amazonaws.com`) that was served by the Regional endpoint in the Europe (Stockholm) Region - eu-north-1. The `endpointType` field value of `regional` indicates that the AWS STS request was served by the global endpoint in the Europe (Stockholm) Region.

```
 {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123456789EXAMPLE:developer",
        "arn": "arn:aws:sts::777788889999:assumed-role/Admin/developer",
        "accountId": "777788889999",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::777788889999:role/Admin",
                "accountId": "777788889999",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2025-02-12T21:44:28Z",
                "mfaAuthenticated": "false"
            }
```

```
        }
    },
    "eventTime": "2025-02-12T22:16:30Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "eu-north-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/2.15.33 Python/3.11.8 Linux/5.10.233-204.894.amzn2int.x86_64 exe/x86_64.amzn.2 prompt/
    "requestParameters": {
        "roleArn": "arn:aws:iam::777788889999:role/test-role",
        "roleSessionName": "test-global-assume-role"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": " <encoded session token blob> ",
            "expiration": "Feb 12, 2025, 11:16:30 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROA987654321EXAMPLE:test-global-assume-role",
            "arn": "arn:aws:sts::777788889999:assumed-role/test-role/test-global-assume-role"
        }
    },
    "additionalEventData": {
        "RequestDetails": {
            "endpointType": "regional",
            "awsServingRegion": "eu-north-1"
        }
    },
    "requestID": "EXAMPLEd-2116-4cd7-bd72-9f72fEXAMPLE",
    "eventID": "EXAMPLEd-219a-48ed-bc54-00e3cEXAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "777788889999",
            "type": "AWS::IAM::Role",
            "ARN": "arn:aws:iam::777788889999:role/test-role"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "777788889999",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "sts.eu-north-1.amazonaws.com"
```

```
    }
}
```

Example sign-in events in CloudTrail log

CloudTrail log files contain events that are formatted using JSON. A sign-in event represents a single sign-in request and includes information about the sign-in principal, the Region, and the date and time of the action.

Example sign-in success event in CloudTrail log file

The following example shows a CloudTrail log entry for a successful sign-in event.

```
{
 "eventVersion": "1.05",
 "userIdentity": {
   "type": "IAMUser",
   "principalId": "AIDACKCEVSQ6C2EXAMPLE",
   "arn":"arn:aws:iam::111122223333:user/John",
   "accountId": "111122223333",
   "userName": "John"
 },
 "eventTime": "2014-07-16T15:49:27Z",
 "eventSource": "signin.amazonaws.com",
 "eventName": "ConsoleLogin",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "192.0.2.110",
 "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",
 "requestParameters": null,
 "responseElements": {
   "ConsoleLogin": "Success"
 },
 "additionalEventData": {
   "MobileVersion": "No",
   "LoginTo": "https://console.aws.amazon.com/s3/",
   "MFAUsed": "No"
 },
 "eventID": "3fcfb182-98f8-4744-bd45-10a395ab61cb"
}
```

The following example shows a CloudTrail log entry for a successful authorization code request.

```
{
"eventVersion": "1.11",
"userIdentity": {
```

```
"type": "AssumedRole",
"principalId": "AROATJHQDX737YZP72NTF:thesjain-Isengard",
"arn": "arn:aws:sts::225989345271:assumed-role/Admin/thesjain-Isengard",
"accountId": "225989345271",
"sessionContext": {
"sessionIssuer": {
"type": "Role",
"principalId": "AROATJHQDX737YZP72NTF",
"arn": "arn:aws:iam::225989345271:role/Admin",
"accountId": "225989345271",
"userName": "Admin"
},
"attributes": {
"creationDate": "2025-11-17T22:50:14Z",
"mfaAuthenticated": "false"
}
}
},
"eventTime": "2025-11-17T22:51:32Z",
"eventSource": "signin.amazonaws.com",
"eventName": "AuthorizeOAuth2Access",
"awsRegion": "us-east-1",
"sourceIPAddress": "52.94.133.136",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.
"requestParameters": {
"scope": "openid",
"redirect_uri": "http://127.0.0.1:53037/oauth/callback",
"code_challenge_method": "SHA-256",
"client_id": "arn:aws:signin:::devtools/same-device"
},
"responseElements": null,
"additionalEventData": {
"success": "true",
"x-amzn-vpce-id": ""
},
"requestID": "e2854c76-1cba-4360-9fd1-5037b591466b",
"eventID": "59e1720d-3deb-44ff-933d-6828be2a860a",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "225989345271",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.3",
"cipherSuite": "TLS_AES_128_GCM_SHA256",
"clientProvidedHostHeader": "us-east-1.signin.aws.amazon.com"
}
```

```
}
```

The following example shows a CloudTrail log entry for a successful OAuth2 token creation request.


```
{
"eventVersion": "1.11",
"userIdentity": {
"type": "AssumedRole",
"principalId": "AROATJHQDX737YZP72NTF:jacobjoj-Isengard",
"arn": "arn:aws:sts::225989345271:assumed-role/Admin/jacobjoj-Isengard",
"accountId": "225989345271",
"sessionContext": {
"sessionIssuer": {
"type": "Role",
"principalId": "AROATJHQDX737YZP72NTF",
"arn": "arn:aws:iam::225989345271:role/Admin",
"accountId": "225989345271",
"userName": "Admin"
},
"attributes": {
"creationDate": "2025-11-18T20:38:10Z",
"mfaAuthenticated": "false"
}
}
},
"eventTime": "2025-11-18T20:38:44Z",
"eventSource": "signin.amazonaws.com",
"eventName": "CreateOAuth2Token",
"awsRegion": "us-east-1",
"sourceIPAddress": "15.248.6.6",
"userAgent": "aws-cli/2.32.0 md/awscrt#0.28.4 ua/2.1 os/macos#24.6.0 md/arch#arm64 lang/python#3.13.9 md/pyimpl
"requestParameters": {
"client_id": "arn:aws:signin:::devtools/same-device"
},
"responseElements": null,
"additionalEventData": {
"success": "true",
"x-amzn-vpce-id": ""
},
"requestID": "94562943-c85b-4dc1-bf72-43b0fd42d6de",
"eventID": "0b338fac-6a10-4740-b34d-1bb6923e799e",
"readOnly": true,
"eventType": "AwsApiCall",
```

```
"managementEvent": true,
"recipientAccountId": "225989345271",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.3",
"cipherSuite": "TLS_AES_128_GCM_SHA256",
"clientProvidedHostHeader": "us-east-1.signin.aws.amazon.com"
}
}
```

For more details about the information contained in CloudTrail log files, see CloudTrail Event Reference in the *AWS CloudTrail User Guide*.

Example sign-in failure event in CloudTrail log file

The following example shows a CloudTrail log entry for a failed sign-in event.

```
{
 "eventVersion": "1.05",
 "userIdentity": {
   "type": "IAMUser",
   "principalId": "AIDACKCEVSQ6C2EXAMPLE",
   "arn":"arn:aws:iam::111122223333:user/JaneDoe",
   "accountId": "111122223333",
   "userName": "JaneDoe"
 },
 "eventTime": "2014-07-08T17:35:27Z",
 "eventSource": "signin.amazonaws.com",
 "eventName": "ConsoleLogin",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "192.0.2.100",
 "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",
 "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"
}
```

From this information, you can determine that the sign-in attempt was made by an IAM user named `JaneDoe`, as shown in the `userIdentity` element. You can also see that the sign-in attempt failed, as shown in the `responseElements` element. You can see that `JaneDoe` tried to sign in to the Amazon SNS console at 5:35 PM (UTC) on July 8, 2014.

Example sign-in failure event caused by incorrect user name

The following example shows a CloudTrail log entry for an unsuccessful sign-in event caused by the user entering an incorrect user name. AWS masks the `userName` text with `HIDDEN_DUE_TO_SECURITY_REASONS` to help prevent exposing potentially sensitive information.


```
{
 "eventVersion": "1.05",
 "userIdentity": {
   "type": "IAMUser",
   "accountId": "123456789012",
   "accessKeyId": "",
   "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
 },
 "eventTime": "2015-03-31T22:20:42Z",
 "eventSource": "signin.amazonaws.com",
 "eventName": "ConsoleLogin",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "192.0.2.101",
 "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0",
 "errorMessage": "No username found in supplied account",
 "requestParameters": null,
 "responseElements": {
   "ConsoleLogin": "Failure"
 },
 "additionalEventData": {
   "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs%23&isauthcode=true",
   "MobileVersion": "No",
   "MFAUsed": "No"
 },
 "eventID": "a7654656-0417-45c6-9386-ea8231385051",
 "eventType": "AwsConsoleSignin",
 "recipientAccountId": "123456789012"
}
```


IAM role trust policy behavior

On September 21st, 2022, AWS made changes to IAM role trust policy behavior to require explicit allows in a role trust policy when a role assumes itself. IAM roles in the legacy behavior allow list have an additionalEventData field present for explicitTrustGrant for `AssumeRole` events. The value of

`explicitTrustGrant` is false when a role on the legacy allow list assumes itself using the legacy behavior. When a role on the legacy allow list assumes itself but the role trust policy behavior has been updated to explicitly allow the role to assume itself, the value of `explicitTrustGrant` is true.

Only a very small number of IAM roles are on the allow list for the legacy behavior, and this field is only present in CloudTrail logs for these roles when they assume themselves. In most cases, it is not necessary for an IAM role to assume itself. AWS recommends updating your processes, code, or configurations to remove this behavior or updating your role trust policies to explicitly allow for this behavior. For more information, see Announcing an update to IAM role trust policy behavior.
