Configuring Amazon S3 security settings and access controls

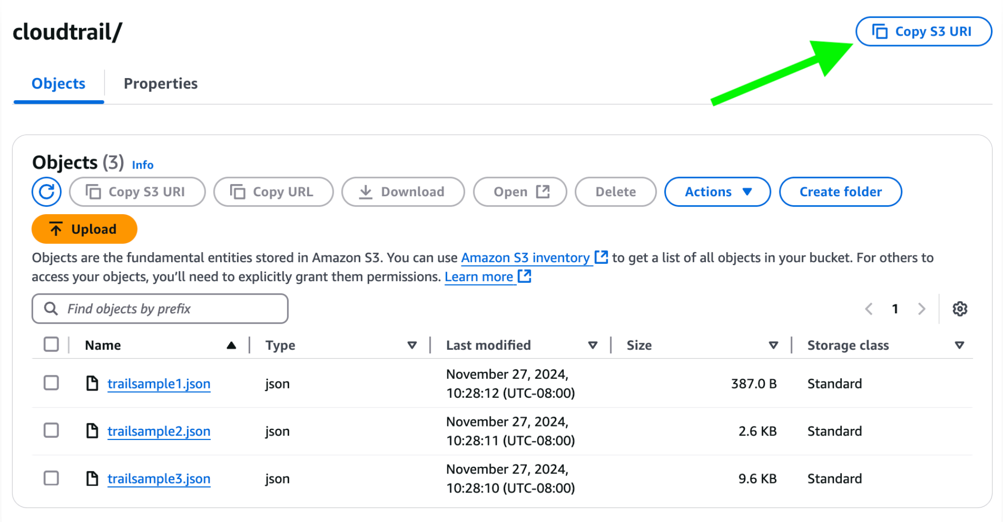
Getting started at an AWS hosted workshop
▼ S3 Security Best Practices
▼ Prepare Your Lab
Attach IAM Role to EC2 Instance
Connect to the EC2 Instance
Bucket Name
▼ Lab 1 - S3 Security Exercises
Require HTTPS
Require SSE-KMS Encryption
Restrict Access to an S3 VPC Endpoint
Use AWS Config Rules to Detect a Public Bucket
Use Amazon Access Analyzer for S3
▼ Lab 2 - S3 Access Grants
S3 Access Grants Lab - Initial Setup
Configure S3 Access Grants for IAM user
▼ Lab 3 - Enabling Malware Protection for S3 by using GuardDuty
Enabling Malware Protection for S3 for your bucket
Testing GuardDuty Malware with an object.
▼ Lab 4 - S3 Access Control Lists
Block Public ACLs
Configure S3 Block Public Access
Disable S3 ACLs
▼ Finding S3 access control lists with S3 Inventory
Enabling Amazon S3 Inventory
Setting up Amazon Athena
Use Amazon Athena to query Amazon S3 Inventory and identify objects with ACL elements
▼ Use Amazon Athena to query CloudTrail logs and identify S3 requests that depend on ACLs
Enabling Cloudtrail data events
Setting up Amazon Athena
Lab Summary

Setting up Amazon Athena

From the AWS console in the top search bar, search and select `s3`.

- Click the bucket name starting with `sid-security-xxxxxxxx` and browse to the folder called "cloudtrail".
- You should see several files titled `trailsample`. Click the `Copy S3 URI` in the upper right, we will use this later when creating the Athena table.

You should see a screen similar to:



cloudtrail/

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	trailsample1.json	json	November 27, 2024, 10:28:12 (UTC-08:00)	387.0 B	Standard
□	trailsample2.json	json	November 27, 2024, 10:28:11 (UTC-08:00)	2.6 KB	Standard
□	trailsample3.json	json	November 27, 2024, 10:28:10 (UTC-08:00)	9.6 KB	Standard

From the AWS console in the top search bar, search and select `athena`.

> ⓘ You might see existing tabs from earlier labs, you can either close them out or just click the blue plus sign on the top right to create a new tab.

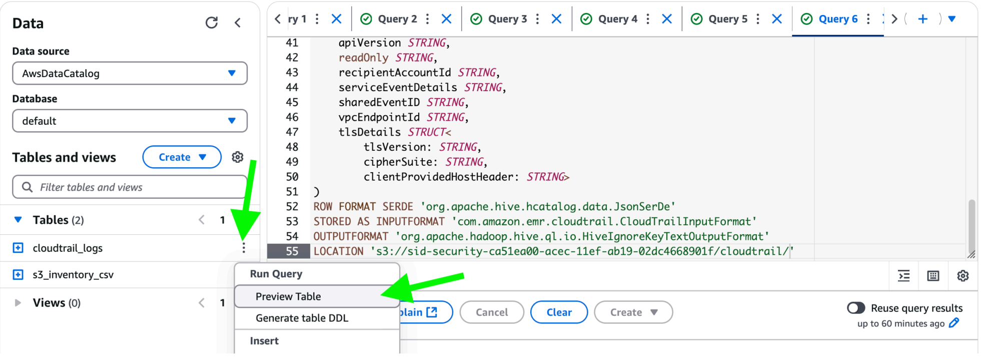
Now we can create our table for CloudTrail logs. We will enter this SQL statement to create a table named cloudtrail_logs in the default Athena database. You must enter the Amazon S3 CloudTrail logs location that we copied earlier in the following LOCATION string with the appropriate values for your configuration.

```
CREATE EXTERNAL TABLE cloudtrail_logs (
    eventVersion STRING,
    userIdentity STRUCT<
        type: STRING,
        principalId: STRING,
        arn: STRING,
        accountId: STRING,
        invokedBy: STRING,
        accessKeyId: STRING,
        userName: STRING,
        sessionContext: STRUCT<
            attributes: STRUCT<
                mfaAuthenticated: STRING,
                creationDate: STRING>,
            sessionIssuer: STRUCT<
                type: STRING,
                principalId: STRING,
                arn: STRING,
                accountId: STRING,
                username: STRING>,
            ec2RoleDelivery: STRING,
            webIdFederationData: MAP<STRING,STRING>>>,
    eventTime STRING,
    eventSource STRING,
    eventName STRING,
    awsRegion STRING,
    sourceIpAddress STRING,
    userAgent STRING,
    errorCode STRING,
    errorMessage STRING,
    requestParameters STRING,
    responseElements STRING,
    additionalEventData STRING,
    requestId STRING,
    eventId STRING,
    resources ARRAY<STRUCT<
        arn: STRING,
        accountId: STRING,
        type: STRING>>,
    eventType STRING,
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventID STRING,
    vpcEndpointId STRING,
    tlsDetails STRUCT<
        tlsVersion: STRING,
        cipherSuite: STRING,
        clientProvidedHostHeader: STRING>
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://sid-security-xxx/cloudtrail/'
TBLPROPERTIES ('classification'='cloudtrail');
```

ⓘ Note that for the LOCATION field, we use the Amazon S3 CloudTrail location we copied previously .

Click Run to create the table. You should see a green "Completed" message at the bottom.

Once the SQL has been run, you should see the `cloudtrail_logs` in your list of tables and can run Athena queries against your CloudTrail logs. Click on the three vertical dots next the the cloudtrail_logs table and click `Preview Table` at the top, as shown in the following figure.



Data source: AwsDataCatalog

Database: default

```
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventID STRING,
    vpcEndpointId STRING,
    tlsDetails STRUCT<
        tlsVersion: STRING,
        cipherSuite: STRING,
        clientProvidedHostHeader: STRING>
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://sid-security-ca51ea00-acec-11ef-ab19-02dc4668901f/cloudtrail/'
```

This will display a preview of the records in the results section, you can scroll back and forth to see the specific fields.

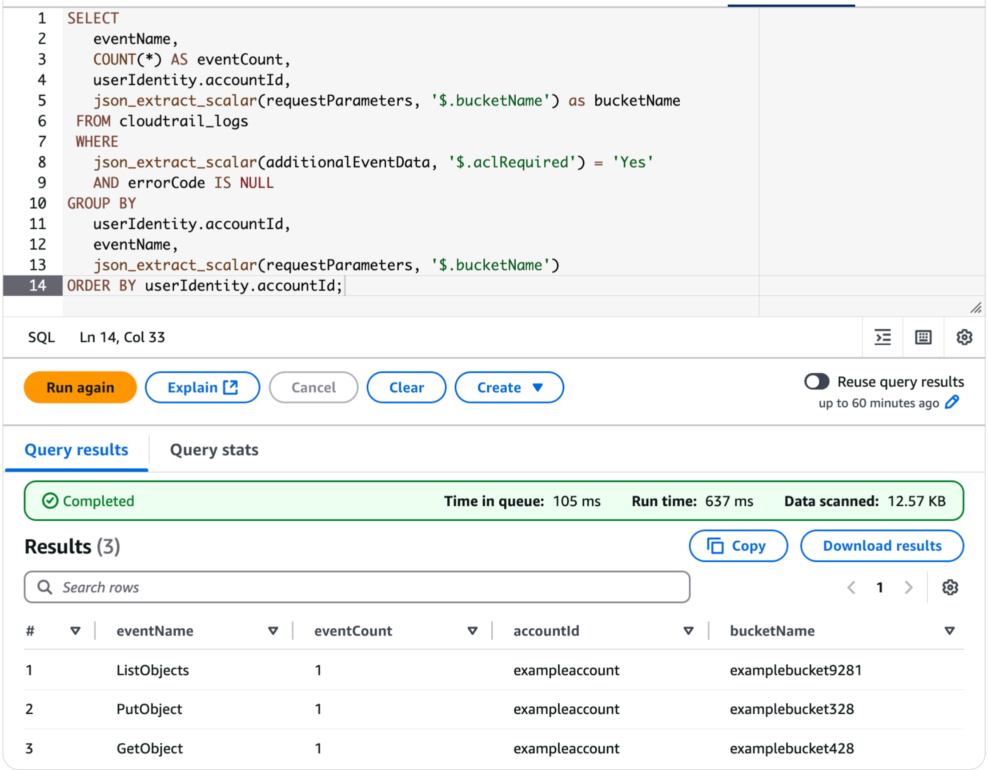
Now that our Athena table is created we can start querying the CloudTrail logs. Click `Clear` at the bottom of the existing query window and copy and paste the following query:

```
SELECT
   eventName,
   COUNT(*) AS eventCount,
   userIdentity.accountId,
   json_extract_scalar(requestParameters, '$.bucketName') as bucketName
 FROM cloudtrail_logs
 WHERE
   json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
   AND errorCode IS NULL
```

```
GROUP BY
    userIdentity.accountId,
    eventName,
    json_extract_scalar(requestParameters, '$.bucketName')
ORDER BY userIdentity.accountId;
```

Before we run this query, lets take a step back and understand why we are doing so. Since we turned on data events in CloudTrail, we are now capturing S3 API calls and specify in the above query we are looking for a relatively new field (as of 2/15/23) called `aclRequired`. The new aclRequired field in Amazon S3 server access logs and AWS CloudTrail logs gives you information on each S3 request to indicate whether or not the request required an ACL for authorization. Its value is either “Yes” or absent in AWS CloudTrail. The purpose of this field is to show you which requests will require a modification to your bucket policy or requests before you can disable ACLs. More simply put this will show active objects that applications or users are using with ACL associated with them.

Now that we have that information go ahead and run the query.



```
SELECT
   eventName,
   COUNT(*) AS eventCount,
   userIdentity.accountId,
   json_extract_scalar(requestParameters, '$.bucketName') as bucketName
 FROM cloudtrail_logs
 WHERE
   json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
   AND errorCode IS NULL
GROUP BY
   userIdentity.accountId,
   eventName,
   json_extract_scalar(requestParameters, '$.bucketName')
ORDER BY userIdentity.accountId;
```

SQL Ln 14, Col 33

Completed Time in queue: 105 ms Run time: 637 ms Data scanned: 12.57 KB

Results (3)

#	eventName	eventCount	accountId	bucketName
1	ListObjects	1	exampleaccount	examplebucket9281
2	PutObject	1	exampleaccount	examplebucket328
3	GetObject	1	exampleaccount	examplebucket428

The output of this query indicates that PutObject, GetObject, and ListBucket requests depend on ACLs to succeed. To be able to follow the security best practice of disabling ACLs, we need to allow access to these users/applications in the bucket policy and verify ACLs are no longer being set on requests.

You might come up with an empty result from this query. This means that during this time period, no requests would have been rejected with ACLs disabled.

You should now be able to use CloudTrail logs to identify S3 Objects that are actively using ACLs.

✓ **Lab complete**
Congratulations you have completed the S3 Access Control Lists (ACLs) labs. In this lab, you learned how to manage your S3 ACL permissions, detect where they may still be present in your AWS accounts, and disable ACLs on your buckets.

Previous