/ Knowledge Center / How do I analyze my Amazon S3 server access logs using Athena?

How do I analyze my Amazon S3 server access logs using Athena?

6 minute read

1

FEEDBACK

How do I query Amazon Simple Storage Service (Amazon S3) server access logs in Amazon Athena?

Resolution

Amazon S3 stores server access logs as objects in an S3 bucket. You can use Athena to quickly analyze and query server access logs.

1. Turn on server access logging for your S3 bucket, if you haven't already. Note the values for **Target bucket** and **Target prefix**—you need both to specify the Amazon S3 location in an Athena query.

2. Open the Amazon Athena console.

3. In the **Query editor**, run a DDL statement to create a database:
Note: It's a best practice to create the database in the same AWS Region as your S3 bucket.

```
create database s3_access_logs_db
```

4. Create a table schema in the database. In the following example, the **STRING** and **BIGINT** data type values are the access log properties. You can query these properties in Athena. For **LOCATION**, enter the S3 bucket and prefix path from Step 1. Make sure to include a forward slash (/) at the end of the prefix (for example, **s3://doc-example-bucket/prefix/**). If you're not using a prefix, then include a forward slash (/) at the end of the bucket name (for example, **s3://doc-example-bucket/**):

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://awsexamplebucket1-logs/prefix/'
```

5. In the left pane, under **Tables**, choose **Preview table** from the menu button that's next to the table name. If you see data from the server access logs in the **Results** window (such as **bucketowner**, **bucket**, and **requestdatetime**), you successfully created the Athena table. You can now query the Amazon S3 server access logs.

Example queries

To find the request for a deleted object, use the following query:

```
SELECT * FROM s3_access_logs_db.mybucket_logs WHERE
key = 'images/picture.jpg' AND operation like '%DELETE%';
```

To show Amazon S3 request IDs for requests that resulted in **403 Access Denied** errors, use the following query:

```
SELECT requestdatetime, requester, operation, requestid, hostid FROM s3_access_log
WHERE httpstatus = '403';
```

To find Amazon S3 request IDs for **HTTP 5xx** errors in a specific time period (including key and error code), run the following query:

```
SELECT requestdatetime, key, httpstatus, errorcode, requestid, hostid FROM s3_acce
WHERE httpstatus like '5%' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-09-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-09-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

To show who deleted an object and when, including the timestamp, IP address, and AWS Identity and Access Management (IAM) role, use the following query:

```
SELECT requestdatetime, remoteip, requester, key FROM s3_access_logs_db.mybucket_l
key = 'images/picture.jpg' AND operation like '%DELETE%';
```

To show all operations performed by an IAM role, use the following query:

```
SELECT * FROM s3_access_logs_db.mybucket_logs WHERE
requester='arn:aws:iam::123456789123:user/user_name';
```

To show all operations performed on an object in a specific time period, use the following query:

```
SELECT SUM(bytessent) as uploadtotal,
SUM(objectsize) as downloadtotal,
SUM(bytessent + objectsize) AS total FROM s3_access_logs_db.mybucket_logs
WHERE remoteIP='1.2.3.4' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01','yyyy-MM-dd')
AND parse_datetime('2021-08-01','yyyy-MM-dd');
```

To show how much data was transferred through an IP address during a specific time period, use the following query:

```
SELECT SUM(bytessent) as uploadtotal,
SUM(objectsize) as downloadtotal,
SUM(bytessent + objectsize) AS total FROM s3_access_logs_db.mybucket_logs
WHERE remoteIP='1.2.3.4' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01','yyyy-MM-dd')
AND parse_datetime('2021-08-01','yyyy-MM-dd');
```

To show all expire operations performed by lifecycle rules in a specific time period, use the following query:

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE operation = 'S3.EXPIRE.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-09-18:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-09-19:00:00:00','yyyy-MM-dd:HH:mm:ss');
```

To count the number of objects expired in a specific time period, use the following query:

```
SELECT count(*) as ExpireCount
FROM s3_access_logs_db.mybucket_logs
WHERE operation = 'S3.EXPIRE.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-09-18:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-09-19:00:00:00','yyyy-MM-dd:HH:mm:ss');
```

To show all transition operations performed by lifecycle rules in a specific time period, use the following query:

```
SELECT * FROM s3_access_logs_db.mybucket_logs
WHERE operation like 'S3.TRANSITION%' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-09-18:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-09-19:00:00:00','yyyy-MM-dd:HH:mm:ss');
```

To show all requesters grouped by Signature Version, use the following query:

```
SELECT requester, Sigv, Count(Sigv) as SigCount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, Sigv;
```

To show all anonymous requesters that are making requests in a specific time period, use the following query:

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db.mybucket_logs
WHERE Requester IS NULL AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

To show all requesters that are sending PUT object requests in a specific time period, use the following query:

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.PUT.OBJECT' AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

To show all requesters that are sending GET object requests in a specific time period, use the following query:

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.GET.OBJECT' AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

To show all anonymous requesters that are making requests in a specific time period, use the following query:

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db.mybucket_logs
WHERE Requester IS NULL AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2021-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

To show all requesters (ordered by highest turnaround time in a specific time period), use the following query:

```
SELECT * FROM s3_access_logs_db.mybucket_logs
NOT turnaroundtime='-' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2021-09-18:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND
```

```
parse_datetime('2021-09-19:00:00:00','yyyy-MM-dd:HH:mm:ss')
ORDER BY CAST(turnaroundtime AS INT) DESC;
```

It's a best practice to create a lifecycle policy for your server access logs bucket. Configure the lifecycle policy to periodically remove log files. This reduces the amount of data that Athena analyzes for each query.

Related information

Analyzing Amazon S3 server access logs using Amazon OpenSearch Service

Amazon S3 server access log format

Querying AWS service logs

Follow

Topics

Analytics

Tags

Amazon Athena

Language

English

Related videos

Watch Bukola's video to learn more (5:43)

AWS OFFICIAL

Updated a year ago

No comments

Comment on this article

Preview | Formatting guide

Start writing your comment

Clear

Post comment

Relevant content

S3 server access logs to Cloudwatch?

ACCEPTED ANSWER

AWS-User-2181865
asked 5 years ago

partitioning s3 access logs to optimize athena queries

AWS-User-7164347
asked a year ago

S3 Server Access Logging - Another Account

Deivis-Campos-0022
asked a year ago

Avoid recursive S3 server access logging + TrustedAdvisor warning

AWS-User-8416516

asked a year ago

Enable S3 server access logging - Target account is a cross account

ACCEPTED ANSWER

rePost-User-6620266
asked 3 months ago

How do I analyze my Application Load Balancer access logs using Amazon Athena?

AWS OFFICIAL
Updated a month ago

Why aren't my Amazon S3 server access logs getting delivered?

AWS OFFICIAL
Updated 5 months ago

How do I analyze the Amazon VPC flow logs using Amazon Athena?

AWS OFFICIAL
Updated 10 months ago

How can I audit deleted or missing objects from my Amazon S3 bucket?

AWS OFFICIAL
Updated a year ago

Troubleshoot 403 Access Denied error in Amazon S3

EXPERT
Gayathri Krishnamoort...
published 3 months ago