NetApp

Audit log file format

StorageGRID 11.7

NetApp
March 05, 2024

Table of Contents

Audit log file format

Audit log file format: Overview

The audit log files are found on every Admin Node and contain a collection of individual audit messages.

Each audit message contains the following:

- The Coordinated Universal Time (UTC) of the event that triggered the audit message (ATIM) in ISO 8601 format, followed by a space:

  YYYY-MM-DDTHH:MM:SS.UUUUUU, where *UUUUUU* are microseconds.

- The audit message itself, enclosed within square brackets and beginning with `AUDT`.

The following example shows three audit messages in an audit log file (line breaks added for readability). These messages were generated when a tenant created an S3 bucket and added two objects to that bucket.

```
2019-08-07T18:43:30.247711
[AUDT:[RSLT(FC32):SUCS][CNID(UI64):1565149504991681][TIME(UI64):73520][SAI
P(IPAD):"10.224.2.255"][S3AI(CSTR):"17530064241597054718"]
[SACC(CSTR):"s3tenant"][S3AK(CSTR):"SGKH9100SCkNB8M3MTWNt-
PhoTDwB9JOk7PtyLkQmA=="][SUSR(CSTR):"urn:sgws:identity::175300642415970547
18:root"]
[SBAI(CSTR):"17530064241597054718"][SBAC(CSTR):"s3tenant"][S3BK(CSTR):"buc
ket1"][AVER(UI32):10][ATIM(UI64):1565203410247711]
[ATYP(FC32):SPUT][ANID(UI32):12454421][AMID(FC32):S3RQ][ATID(UI64):7074142
142472611085]]

2019-08-07T18:43:30.783597
[AUDT:[RSLT(FC32):SUCS][CNID(UI64):1565149504991696][TIME(UI64):120713][SA
IP(IPAD):"10.224.2.255"][S3AI(CSTR):"17530064241597054718"]
[SACC(CSTR):"s3tenant"][S3AK(CSTR):"SGKH9100SCkNB8M3MTWNt-
PhoTDwB9JOk7PtyLkQmA=="][SUSR(CSTR):"urn:sgws:identity::175300642415970547
18:root"]
[SBAI(CSTR):"17530064241597054718"][SBAC(CSTR):"s3tenant"][S3BK(CSTR):"buc
ket1"][S3KY(CSTR):"fh-small-0"]
[CBID(UI64):0x779557A069B2C037][UUID(CSTR):"94BA6949-38E1-4B0C-BC80-
EB44FB4FCC7F"][CSIZ(UI64):1024][AVER(UI32):10]
[ATIM(UI64):1565203410783597][ATYP(FC32):SPUT][ANID(UI32):12454421][AMID(F
C32):S3RQ][ATID(UI64):8439606722108456022]]

2019-08-07T18:43:30.784558
[AUDT:[RSLT(FC32):SUCS][CNID(UI64):1565149504991693][TIME(UI64):121666][SA
IP(IPAD):"10.224.2.255"][S3AI(CSTR):"17530064241597054718"]
[SACC(CSTR):"s3tenant"][S3AK(CSTR):"SGKH9100SCkNB8M3MTWNt-
PhoTDwB9JOk7PtyLkQmA=="][SUSR(CSTR):"urn:sgws:identity::175300642415970547
18:root"]
[SBAI(CSTR):"17530064241597054718"][SBAC(CSTR):"s3tenant"][S3BK(CSTR):"buc
ket1"][S3KY(CSTR):"fh-small-2000"]
[CBID(UI64):0x180CBD8E678EED17][UUID(CSTR):"19CE06D0-D2CF-4B03-9C38-
E578D66F7ADD"][CSIZ(UI64):1024][AVER(UI32):10]
[ATIM(UI64):1565203410784558][ATYP(FC32):SPUT][ANID(UI32):12454421][AMID(F
C32):S3RQ][ATID(UI64):13489590586043706682]]
```

In their default format, the audit messages in the audit log files aren't easy to read or interpret. You can use the audit-explain tool to obtain simplified summaries of the audit messages in the audit log. You can use the audit-sum tool to summarize how many write, read, and delete operations were logged and how long these operations took.

Use audit-explain tool

You can use the `audit-explain` tool to translate the audit messages in the audit log in

to an easy-to-read format.

Before you begin

- You must have specific access permissions.

- You must have the `Passwords.txt` file.

- You must know the IP address of the primary Admin Node.

About this task

The `audit-explain` tool, available on the primary Admin Node, provides simplified summaries of the audit messages in an audit log.

> ⓘ The `audit-explain` tool is primarily intended for use by technical support during troubleshooting operations. Processing `audit-explain` queries can consume a large amount of CPU power, which might impact StorageGRID operations.

This example shows typical output from the `audit-explain` tool. These four SPUT audit messages were generated when the S3 tenant with account ID 92484777680322627870 used S3 PUT requests to create a bucket named "bucket1" and add three objects to that bucket.

```
SPUT S3 PUT bucket bucket1 account:92484777680322627870 usec:124673
SPUT S3 PUT object bucket1/part1.txt tenant:92484777680322627870
cbid:9DCB157394F99FE5 usec:101485
SPUT S3 PUT object bucket1/part2.txt tenant:92484777680322627870
cbid:3CFBB07AB3D32CA9 usec:102804
SPUT S3 PUT object bucket1/part3.txt tenant:92484777680322627870
cbid:5373D73831ECC743 usec:93874
```

The `audit-explain` tool can do the following:

- Process plain or compressed audit logs. For example:

  `audit-explain audit.log`

  `audit-explain 2019-08-12.txt.gz`

- Process multiple files simultaneously. For example:

  `audit-explain audit.log 2019-08-12.txt.gz 2019-08-13.txt.gz`

  `audit-explain /var/local/audit/export/*`

- Accept input from a pipe, which allows you to filter and preprocess the input using the `grep` command or other means. For example:

  `grep SPUT audit.log | audit-explain`

  `grep bucket-name audit.log | audit-explain`

Because audit logs can be very large and slow to parse, you can save time by filtering parts that you want to look at and running `audit-explain` on the parts, instead of the entire file.

> (i) The `audit-explain` tool does not accept compressed files as piped input. To process compressed files, provide their file names as command-line arguments, or use the `zcat` tool to decompress the files first. For example:
>
> `zcat audit.log.gz | audit-explain`

Use the `help` `(-h)` option to see the available options. For example:

`$ audit-explain -h`

Steps

1. Log in to the primary Admin Node:

   a. Enter the following command: `ssh admin@`*`primary_Admin_Node_IP`*

   b. Enter the password listed in the `Passwords.txt` file.

   c. Enter the following command to switch to root: `su -`

   d. Enter the password listed in the `Passwords.txt` file.

   When you are logged in as root, the prompt changes from `$` to `#`.

2. Enter the following command, where `/var/local/audit/export/audit.log` represents the name and the location of the file or files you want to analyze:

   `$ audit-explain /var/local/audit/export/audit.log`

   The `audit-explain` tool prints human-readable interpretations of all messages in the specified file or files.

   > (i) To reduce line lengths and to aid readability, timestamps aren't shown by default. If you want to see the timestamps, use the timestamp (`-t`) option.

Use audit-sum tool

You can use the `audit-sum` tool to count the write, read, head, and delete audit messages and to see the minimum, maximum, and average time (or size) for each operation type.

Before you begin

- You must have specific access permissions.
- You must have the `Passwords.txt` file.
- You must know the IP address of the primary Admin Node.

About this task

The `audit-sum` tool, available on the primary Admin Node, summarizes how many write, read, and delete

operations were logged and how long these operations took.

> ℹ️ The `audit-sum` tool is primarily intended for use by technical support during troubleshooting operations. Processing `audit-sum` queries can consume a large amount of CPU power, which might impact StorageGRID operations.

This example shows typical output from the `audit-sum` tool. This example shows how long protocol operations took.

```
   message group          count      min(sec)          max(sec)
average(sec)
   =============          =====      ========          ========
============
   IDEL                     274
   SDEL                  213371        0.004            20.934
0.352
   SGET                  201906        0.010          1740.290
1.132
   SHEA                   22716        0.005             2.349
0.272
   SPUT                 1771398        0.011          1770.563
0.487
```

The `audit-sum` tool provides counts and times for the following S3, Swift, and ILM audit messages in an audit log:

Code	Description	Refer to
ARCT	Archive Retrieve from Cloud-Tier	ARCT: Archive Retrieve from Cloud-Tier
ASCT	Archive Store Cloud-Tier	ASCT: Archive Store Cloud-Tier
IDEL	ILM Initiated Delete: Logs when ILM starts the process of deleting an object.	IDEL: ILM Initiated Delete
SDEL	S3 DELETE: Logs a successful transaction to delete an object or bucket.	SDEL: S3 DELETE
SGET	S3 GET: Logs a successful transaction to retrieve an object or list the objects in a bucket.	SGET: S3 GET
SHEA	S3 HEAD: Logs a successful transaction to check for the existence of an object or bucket.	SHEA: S3 HEAD
SPUT	S3 PUT: Logs a successful transaction to create a new object or bucket.	SPUT: S3 PUT

Code	Description	Refer to
WDEL	Swift DELETE: Logs a successful transaction to delete an object or container.	WDEL: Swift DELETE
WGET	Swift GET: Logs a successful transaction to retrieve an object or list the objects in a container.	WGET: Swift GET
WHEA	Swift HEAD: Logs a successful transaction to check for the existence of an object or container.	WHEA: Swift HEAD
WPUT	Swift PUT: Logs a successful transaction to create a new object or container.	WPUT: Swift PUT

The `audit-sum` tool can do the following:

- Process plain or compressed audit logs. For example:

  `audit-sum audit.log`

  `audit-sum 2019-08-12.txt.gz`

- Process multiple files simultaneously. For example:

  `audit-sum audit.log 2019-08-12.txt.gz 2019-08-13.txt.gz`

  `audit-sum /var/local/audit/export/*`

- Accept input from a pipe, which allows you to filter and preprocess the input using the `grep` command or other means. For example:

  `grep WGET audit.log | audit-sum`

  `grep bucket1 audit.log | audit-sum`

  `grep SPUT audit.log | grep bucket1 | audit-sum`

> ⓘ This tool does not accept compressed files as piped input. To process compressed files, provide their file names as command-line arguments, or use the `zcat` tool to decompress the files first. For example:
>
>   `audit-sum audit.log.gz`
>
>   `zcat audit.log.gz | audit-sum`

You can use command-line options to summarize operations on buckets separately from operations on objects or to group message summaries by bucket name, by time period, or by target type. By default, the summaries show the minimum, maximum, and average operation time, but you can use the `size (-s)` option to look at

object size instead.

Use the `help (-h)` option to see the available options. For example:

```
$ audit-sum -h
```

Steps

1. Log in to the primary Admin Node:

    a. Enter the following command: `ssh admin@`*`primary_Admin_Node_IP`*

    b. Enter the password listed in the `Passwords.txt` file.

    c. Enter the following command to switch to root: `su -`

    d. Enter the password listed in the `Passwords.txt` file.

    When you are logged in as root, the prompt changes from `$` to `#`.

2. If you want to analyze all messages related to write, read, head, and delete operations, follow these steps:

    a. Enter the following command, where `/var/local/audit/export/audit.log` represents the name and the location of the file or files you want to analyze:

    ```
    $ audit-sum /var/local/audit/export/audit.log
    ```

    This example shows typical output from the `audit-sum` tool. This example shows how long protocol operations took.

message group	count	min(sec)	max(sec)	average(sec)
=============	=====	========	========	============
IDEL	274			
SDEL	213371	0.004	20.934	0.352
SGET	201906	0.010	1740.290	1.132
SHEA	22716	0.005	2.349	0.272
SPUT	1771398	0.011	1770.563	0.487

    In this example, SGET (S3 GET) operations are the slowest on average at 1.13 seconds, but SGET and SPUT (S3 PUT) operations both show long worst-case times of about 1,770 seconds.

    b. To show the slowest 10 retrieval operations, use the grep command to select only SGET messages and add the long output option (`-l`) to include object paths:

    ```
    grep SGET audit.log | audit-sum -l
    ```

    The results include the type (object or bucket) and path, which allows you to grep the audit log for other

messages relating to these particular objects.

```
 Total:            201906 operations
     Slowest:         1740.290 sec
     Average:            1.132 sec
     Fastest:            0.010 sec
     Slowest operations:
         time(usec)         source ip         type       size(B) path
         ==========  ================  ============  ============ ====
         1740289662    10.96.101.125        object    5663711385
backup/r9O1OaQ8JB-1566861764-4519.iso
         1624414429    10.96.101.125        object    5375001556
backup/r9O1OaQ8JB-1566861764-6618.iso
         1533143793    10.96.101.125        object    5183661466
backup/r9O1OaQ8JB-1566861764-4518.iso
             70839    10.96.101.125        object         28338
bucket3/dat.1566861764-6619
             68487    10.96.101.125        object         27890
bucket3/dat.1566861764-6615
             67798    10.96.101.125        object         27671
bucket5/dat.1566861764-6617
             67027    10.96.101.125        object         27230
bucket5/dat.1566861764-4517
             60922    10.96.101.125        object         26118
bucket3/dat.1566861764-4520
             35588    10.96.101.125        object         11311
bucket3/dat.1566861764-6616
             23897    10.96.101.125        object         10692
bucket3/dat.1566861764-4516
```

From this example output, you can see that the three slowest S3 GET requests were for objects about 5 GB in size, which is much larger than the other objects. The large size accounts for the slow worst-case retrieval times.

3. If you want to determine what sizes of objects are being ingested into and retrieved from your grid, use the size option (-s):

```
audit-sum -s audit.log
```

```
   message group            count         min(MB)            max(MB)
average(MB)
   =============            =====        ========           ========
============
   IDEL                       274          0.004           5000.000
1654.502
   SDEL                    213371          0.000             10.504
1.695
   SGET                    201906          0.000           5000.000
14.920
   SHEA                     22716          0.001             10.504
2.967
   SPUT                   1771398          0.000           5000.000
2.495
```

In this example, the average object size for SPUT is under 2.5 MB, but the average size for SGET is much larger. The number of SPUT messages is much higher than the number of SGET messages, indicating that most objects are never retrieved.

4. If you want to determine if retrievals were slow yesterday:

   a. Issue the command on the appropriate audit log and use the group-by-time option (-gt), followed by the time period (for example, 15M, 1H, 10S):

   ```
   grep SGET audit.log | audit-sum -gt 1H
   ```

```
   message group             count     min(sec)          max(sec)
average(sec)
   =============             =====     ========          ========
============
   2019-09-05T00              7591        0.010          1481.867
1.254
   2019-09-05T01              4173        0.011          1740.290
1.115
   2019-09-05T02             20142        0.011          1274.961
1.562
   2019-09-05T03             57591        0.010          1383.867
1.254
   2019-09-05T04            124171        0.013          1740.290
1.405
   2019-09-05T05            420182        0.021          1274.511
1.562
   2019-09-05T06           1220371        0.015          6274.961
5.562
   2019-09-05T07            527142        0.011          1974.228
2.002
   2019-09-05T08            384173        0.012          1740.290
1.105
   2019-09-05T09             27591        0.010          1481.867
1.354
```

These results show that S3 GET traffic spiked between 06:00 and 07:00. The max and average times are both considerably higher at these times as well, and they did not ramp up gradually as the count increased. This suggests that capacity was exceeded somewhere, perhaps in the network or in the grid's ability to process requests.

b. To determine what size objects were being retrieved each hour yesterday, add the size option (-s) to the command:

```
grep SGET audit.log | audit-sum -gt 1H -s
```

```
   message group              count       min(B)           max(B)
average(B)
   =============            =====     ========          ========
============
   2019-09-05T00              7591        0.040         1481.867
1.976
   2019-09-05T01              4173        0.043         1740.290
2.062
   2019-09-05T02             20142        0.083         1274.961
2.303
   2019-09-05T03             57591        0.912         1383.867
1.182
   2019-09-05T04            124171        0.730         1740.290
1.528
   2019-09-05T05            420182        0.875         4274.511
2.398
   2019-09-05T06           1220371        0.691   5663711385.961
51.328
   2019-09-05T07            527142        0.130         1974.228
2.147
   2019-09-05T08            384173        0.625         1740.290
1.878
   2019-09-05T09             27591        0.689         1481.867
1.354
```

These results indicate that some very large retrievals occurred when the overall retrieval traffic was at its maximum.

c. To see more detail, use the audit-explain tool to review all the SGET operations during that hour:

```
grep 2019-09-05T06 audit.log | grep SGET | audit-explain | less
```

If the output of the grep command is expected to be many lines, add the `less` command to show the contents of the audit log file one page (one screen) at a time.

5. If you want to determine if SPUT operations on buckets are slower than SPUT operations for objects:

a. Start by using the `-go` option, which groups messages for object and bucket operations separately:

```
grep SPUT sample.log | audit-sum -go
```

```
   message group               count      min(sec)        max(sec)
average(sec)
   =============               =====      ========        ========
============
   SPUT.bucket                     1         0.125           0.125
0.125
   SPUT.object                    12         0.025           1.019
0.236
```

The results show that SPUT operations for buckets have different performance characteristics than SPUT operations for objects.

b. To determine which buckets have the slowest SPUT operations, use the -gb option, which groups messages by bucket:

```
grep SPUT audit.log | audit-sum -gb
```

```
   message group                    count      min(sec)        max(sec)
average(sec)
   =============                    =====      ========        ========
============
   SPUT.cho-non-versioning          71943       0.046        1770.563
1.571
   SPUT.cho-versioning              54277       0.047        1736.633
1.415
   SPUT.cho-west-region             80615       0.040          55.557
1.329
   SPUT.ldt002                    1564563       0.011          51.569
0.361
```

c. To determine which buckets have the largest SPUT object size, use both the -gb and the -s options:

```
grep SPUT audit.log | audit-sum -gb -s
```

```
  message group                    count        min(B)            max(B)
average(B)
   ==============                   =====      ========          ========
============
   SPUT.cho-non-versioning          71943        2.097          5000.000
21.672
   SPUT.cho-versioning              54277        2.097          5000.000
21.120
   SPUT.cho-west-region             80615        2.097           800.000
14.433
   SPUT.ldt002                    1564563        0.000           999.972
0.352
```