監査ログファイルの形式

StorageGRID

NetApp
October 03, 2025

目次

監査ログファイルの形式

監査ログファイルはすべての管理ノードに存在し、一連の監査メッセージが格納されています。

各監査メッセージには次の情報が含まれます。

- 監査メッセージ（ ATIM ）をトリガーしたイベントの協定世界時（ UTC ）を ISO 8601 形式で表した値と、末尾のスペース。

  YYYY-MM-DDTHHH.MM:SS.UUUUUUUU 。ここで、 *UUUUUU* はマイクロ秒です。

- 監査メッセージ。監査メッセージは角かっこで囲まれ、先頭は「 AUDT 」です。

次の例は、監査ログファイル内の 3 つの監査メッセージを示しています（読みやすくするために改行しています）。これらのメッセージは、テナントが S3 バケットを作成し、オブジェクトを 2 つバケットに追加したときに生成されました。

```
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::17530064241597054
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::17530064241597054
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::17530064241597054
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]]
```

デフォルトの形式のままでは、監査ログファイル内の監査メッセージを読んで解釈するのは簡単ではありません。「 audit-explain 」ツールを使用すると、監査ログ内の監査メッセージの簡易サマリーを取得できます。「 audit-sum 」ツールを使用して、ログに記録された書き込み、読み取り、および削除操作の数と、それらの操作に要した時間を要約できます。

関連情報

audit-explain ツールを使用してください

audit-sum ツールを使用します

audit-explain ツールを使用してください

「 audit-explain 」ツールを使用すると、監査ログ内の監査メッセージを読みやすい形式に変換できます。

必要なもの

- 特定のアクセス権限が必要です。
- 「 passwords.txt 」ファイルが必要です。
- プライマリ管理ノードの IP アドレスを確認しておく必要があります。

このタスクについて

プライマリ管理ノードで使用可能な「 audit-explain 」ツールは、監査ログ内の監査メッセージの簡単な概要を提供します。

> 「 audit-explain 」ツールは、主にトラブルシューティング作業中にテクニカルサポートが使用することを目的としています。「 audit-explain 」クエリを処理すると、 StorageGRID の動作に影響を与える可能性がある大量の CPU 消費電力が発生する可能性があります。

この例は 'audit-explain' ツールからの一般的な出力を示しています4 つの SPUT 監査メッセージが、アカウント ID が 92484777680322627870 の S3 テナントが S3 PUT 要求を使用して「 bucket1 」という名前のバケットを作成し、バケットにオブジェクトを 3 つ追加したときに生成されました。

```
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
```

「 audit-explain 」ツールは、プレーン形式または圧縮された監査ログを処理できます。例：

```
audit-explain audit.log
```

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

「 audit-explain 」ツールは、複数のファイルを一度に処理することもできます。例：

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

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

最後に 'audit-explain' ツールは 'grep コマンドまたはその他の方法を使用して入力をフィルタリングおよび前処理できるパイプからの入力を受け入れます例：

```
grep SPUT audit.log | audit-explain
```

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

監査ログは非常に大きくなり、解析に時間がかかる可能性があるため、ファイル全体ではなく、参照したい部分をフィルタリングして、その部分に対して「 audit-explain 」を実行することで、時間を節約できます。

「 audit-explain 」ツールは、圧縮ファイルをパイプ付き入力として受け入れません。圧縮ファイルを処理するには、ファイル名をコマンドライン引数として指定するか、「 zcat 」ツールを使用して最初にファイルを解凍します。例：

```
zcat audit.log.gz | audit-explain
```

利用可能なオプションを表示するには 'help (-h) オプションを使用します例：

```
$ audit-explain -h
```

手順

1. プライマリ管理ノードにログインします。
   a. 次のコマンドを入力します。 ssh admin@*primary_Admin_Node_IP*`
   b. 「 passwords.txt 」ファイルに記載されたパスワードを入力します。
2. 次のコマンドを入力しますここで '/var/local/audit/export/audit.log は ' 分析するファイルの名前と場所を表します

   $audit - /var/local/audit/export/audit.log を説明します

   「 audit-explain 」ツールは、指定されたファイル内のすべてのメッセージを、人間が判読できる形式で出力します。

   各行を短く読みやすくするために、デフォルトではタイムスタンプは表示されません。タイムスタンプを表示するには 'timestamp(`-t`) オプションを使用します

関連情報

SPUT ： S3 PUT

audit-sum ツールを使用します

「 audit-sum 」ツールを使用すると、書き込み、読み取り、 HEAD 、および削除の監査メッセージをカウントしたり、各処理タイプの最小、最大、平均時間（またはサイズ）を表示したりできます。

必要なもの

- 特定のアクセス権限が必要です。
- 「 passwords.txt 」ファイルが必要です。
- プライマリ管理ノードの IP アドレスを確認しておく必要があります。

このタスクについて

プライマリ管理ノードで使用可能な「 audit-sum 」ツールには、ログに記録された書き込み、読み取り、削除の処理数と、それらの処理に要した時間が表示されます。

> 「 audit-sum 」ツールは、主にトラブルシューティング作業時にテクニカルサポートが使用することを目的としています。「 audit-sum 」クエリを処理すると、大量の CPU 消費電力が発生し、 StorageGRID の動作に影響を与える可能性があります。

次の例は 'audit-sum ツールからの一般的な出力を示していますこの例は、プロトコル処理に要した時間を示しています。

```
  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
```

「 audit-sum 」ツールは、監査ログ内の次の S3 、 Swift 、および ILM 監査メッセージのカウントと時間を提供します。

コード	説明	を参照してください
ARCT	アーカイブをクラウド階層から取得します	ARCT ：クラウド階層からアーカイブを取得します
▽ SCT 。△	アーカイブストア - クラウド階層	ASCT ：アーカイブストアのクラウド階層

コード	説明	を参照してください
IDEL	ILM Initiated Delete ： ILM がオブジェクトを削除する処理を開始すると記録されます。	IDEL ： ILM Initiated Delete
SDEL	S3 DELETE ：オブジェクトまたはバケットを削除するトランザクションの成功をログに記録します。	SDEL ： S3 DELETE
SGET	S3 GET ：バケット内のオブジェクトを読み出しまたはリストアップするトランザクションの成功をログに記録します。	SGET ： S3 GET
Shea	S3 HEAD ：オブジェクトまたはバケットの存在を確認するトランザクションの成功をログに記録します。	Shea ： S3 ヘッド
SPUT	S3 PUT ：オブジェクトまたはバケットを新規に作成するトランザクションの成功をログに記録します。	SPUT ： S3 PUT
WDEL	Swift DELETE ：オブジェクトまたはコンテナを削除するトランザクションの成功をログに記録します。	WDEL ： Swift の削除
wget	Swift GET ：コンテナ内のオブジェクトを読み出しまたはリストアップするトランザクションの成功をログに記録します。	wget ： Swift GET
WHEA	Swift HEAD ：オブジェクトまたはコンテナの存在を確認するトランザクションの成功をログに記録します。	WHEA ： Swift ヘッド
WPUT	Swift PUT ：オブジェクトまたはコンテナを新規に作成するトランザクションの成功をログに記録します。	WPUT ： Swift PUT

「 audit-sum 」ツールは、プレーン形式または圧縮形式の監査ログを処理できます。例：

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

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

「 audit-sum 」ツールでは、複数のファイルを一度に処理することもできます。例：

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

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

最後に 'audit-sum ツールでは ' パイプからの入力も受け入れられますこれにより 'grep コマンドまたはその他の方法を使用して ' 入力をフィルタリングおよび前処理できます例：

```
grep WGET audit.log | audit-sum
```

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

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

このツールは、圧縮ファイルをパイプ付き入力として受け入れません。圧縮ファイルを処理するには、ファイル名をコマンドライン引数として指定するか、「 zcat 」ツールを使用して最初にファイルを解凍します。例：

```
audit-sum audit.log.gz
```

```
zcat audit.log.gz | audit-sum
```

コマンドラインオプションを使用して、バケットに対する処理をオブジェクトに対する処理とは別にまとめたり、メッセージの概要をバケット名、期間、ターゲットタイプ別にグループ化したりできます。デフォルトでは ' サマリーには最小 ' 最大 ' 平均のオペレーション時間が表示されますが 'size (-s)' オプションを使用してオブジェクト・サイズを表示することもできます

利用可能なオプションを表示するには 'help (-h) オプションを使用します例：

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

手順

1. プライマリ管理ノードにログインします。
   a. 次のコマンドを入力します。 ssh admin@*primary_Admin_Node_IP*`
   b. 「 passwords.txt 」ファイルに記載されたパスワードを入力します。
2. 書き込み、読み取り、 HEAD 、削除の処理に関連するすべてのメッセージを分析するには、次の手順を実行します。

a. 次のコマンドを入力しますここで '/var/local/audit/export/audit.log は ' 分析するファイルの名前と場所を表します

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

次の例は 'audit-sum ツールからの一般的な出力を示していますこの例は、プロトコル処理に要した時間を示しています。

```
  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
```

この例では、平均処理時間では SGET （ S3 GET ）処理が 1.13 秒と最も長い一方で、最大処理時間では SGET 処理と SPUT （ S3 PUT ）処理がどちらも約 1 、 770 秒と一番長くなっています。

b. 最も時間がかかった読み出し処理を 10 回表示するには、 grep コマンドを使用して SGET メッセージだけを選択し、 long 出力オプション（「 -l 」）を追加してオブジェクトパスを含めます。「 grep SGET audit.log | audit-sum -l 」

結果にはタイプ（オブジェクトまたはバケット）とパスが含まれます。この情報を使用して、監査ログを grep してこれらのオブジェクトに関連する他のメッセージを出力できます。

```
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
```

+ この出力例からは、最も時間がかかった 3 個の S3 GET 要求が、他のオブジェクトよりもはるかに大きい約 5GB のオブジェクトに対して実行されたことがわかります。サイズが大きいと、最悪の場合の読み出し時間が長くなります。

3. グリッドに取り込まれ ' グリッドから取得されるオブジェクトのサイズを決定するには ' size オプション (`s') を使用します

```
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
```

この例では、 SPUT の平均オブジェクトサイズは 2.5MB 未満ですが、 SGET の平均サイズははるかに大きいことがわかります。SPUT メッセージの数は SGET メッセージの数よりもはるかに多く、ほとんどのオブジェクトが読み出されていないことを示しています。

4. 昨日の読み出しに時間がかかっていないかどうかを確認するには、次の手順を実行
   a. 適切な監査ログに対してコマンドを実行し、 group-by-time オプション（ `-gt` ）と期間（例： 15M 、 1H 、 10S ）を使用します。問題

```
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
```

上記の結果は、 06 ： 00 と 07 ： 00 の間に S3 GET トラフィックが急増したことを示しています。この時間帯は最大時間と平均時間も大幅に長くなっており、データの増加に伴って徐々に長くなっているわけではありません。このことから、ネットワークまたはグリッドによる要求の処理能力のどこかでキャパシティを超えた可能性があります。

b. どのサイズのオブジェクトが昨日取得されたかを調べるには ' コマンドに size オプション (`-s') を追加します

```
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
```

この結果から、読み出しトラフィックの量が最大に達したときに、非常に大容量の読み出しが発生したことがわかります。

c. より詳細な情報を表示するには 'audit-explain ツールを使用して ' その時間中のすべての SGET 操作を確認します

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

grep コマンドの出力が多くなると予想される場合は、「 less 」コマンドを追加して、監査ログファイルの内容を 1 ページ（ 1 画面）ずつ表示します。

5. バケットに対する SPUT 処理にオブジェクトに対する SPUT 処理よりも時間がかかっているかどうかを確認するには、次の手順を実行します。

a. まず、 -go' オプションを使用します。このオプションは、オブジェクトおよびバケット操作のメッセージを個別にグループ化します。

```
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
```

上記の結果から、バケットに対する SPUT 処理とオブジェクトに対する SPUT 処理でパフォーマンス特性が異なることがわかります。

b. SPUT 処理に最も時間がかかっているバケットを特定するには、バケットごとにメッセージをグループ化する -gb オプションを使用します。

```
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. SPUT オブジェクトのサイズが最も大きいバケットを特定するには '-sGB' オプションと -s オプションの両方を使用します

```
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
```

関連情報

audit-explain ツールを使用してください

著作権に関する情報

商標に関する情報