監査ログファイルの形式

StorageGRID software

NetApp
December 03, 2025

目次

監査ログファイルの形式

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上に存在し、個々の監査メッセージのコレクションが含まれています。

各監査メッセージには次の内容が含まれます。

- 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 形式で示し、その後にスペースを 1 つ入力します。

  YYYY-MM-DDTHH:MM:SS.UUUUUU、 どこ `*UUUUUU*` マイクロ秒です。

- 監査メッセージ自体は角括弧で囲まれ、 `AUDT` 。

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が追加されています)。これらのメッセージは、テナントが S3 バケットを作成し、そのバケットに 2 つのオブジェクトを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]]
```

デフォルトの形式では、監査ログ ファイル内の監査メッセージは読みやすく解釈しにくいものです。使用することができます"監査説明ツール"監査ログ内の監査メッセージの簡略化された要約を取得します。使用することができます"監査合計ツール"記録された書き込み、読み取り、削除操作の数と、これらの操作にかかった時間を要約します。

監査説明ツールを使用する

使用することができます `audit-explain`監査ログ内の監査メッセージを読みやすい形式に

変換するツール。

開始する前に

- あなたが持っている"特定のアクセス権限"。
- あなたは `Passwords.txt`ファイル。
- プライマリ管理ノードの IP アドレスを知っておく必要があります。

タスク概要

その `audit-explain`プライマリ管理ノードで利用可能なツールは、監査ログ内の監査メッセージの簡略化された概要を提供します。

> その `audit-explain`このツールは主に、トラブルシューティング操作中にテクニカル サポートが使用することを目的としています。処理 `audit-explain`クエリは大量の CPU パワーを消費する可能性があり、 StorageGRID の操作に影響を与える可能性があります。

この例は、 `audit-explain`道具。これら4つ"吐き出す"アカウント ID 92484777680322627870 の S3 テナントが S3 PUT リクエストを使用して「bucket1」という名前のバケットを作成し、そのバケットに 3 つのオブジェクトを追加したときに、監査メッセージが生成されました。

```
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
```

その `audit-explain`このツールは次のことができます。

- プレーンまたは圧縮された監査ログを処理します。例えば：

  ```
  audit-explain audit.log

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

- 複数のファイルを同時に処理します。例えば：

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

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

- パイプからの入力を受け入れ、入力をフィルタリングしたり前処理したりすることができます。 `grep`命令またはその他の手段。例えば：

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

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

監査ログは非常に大きく、解析に時間がかかるため、確認したい部分をフィルタリングして実行することで時間を節約できます。 `audit-explain`ファイル全体ではなく、部分ごとに行います。

> その `audit-explain`ツールはパイプ入力として圧縮ファイルを受け入れません。圧縮ファイルを処理するには、コマンドライン引数としてファイル名を指定するか、 `zcat`まずファイルを解凍するツールです。例えば：
>
> ```
> zcat audit.log.gz | audit-explain
> ```

使用 `help (-h)`利用可能なオプションを表示するには、オプションを選択します。例えば：

```
$ 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. 記載されているパスワードを入力してください `Passwords.txt`ファイル。
   c. ルートに切り替えるには、次のコマンドを入力します。 `su -`
   d. 記載されているパスワードを入力してください `Passwords.txt`ファイル。

      ルートとしてログインすると、プロンプトは `$`に `#`。
2. 次のコマンドを入力します。 `/var/local/log/audit.log`分析するファイルの名前と場所を表します。

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

   その `audit-explain`このツールは、指定されたファイル内のすべてのメッセージの人間が読める形式の解釈を出力します。

   > 行の長さを短くし、読みやすくするために、タイムスタンプはデフォルトでは表示されません。タイムスタンプを確認したい場合は、タイムスタンプを使用してください(`-t`) オプション。

監査合計ツールを使用する

使用することができます `audit-sum`書き込み、読み取り、ヘッド、削除の監査メッセージをカウントし、各操作タイプの最小時間、最大時間、平均時間 (またはサイズ) を確認するツール。

開始する前に

- あなたが持っている"特定のアクセス権限"。
- あなたは `Passwords.txt`ファイル。
- プライマリ管理ノードの IP アドレスを知っておく必要があります。

タスク概要

その `audit-sum`プライマリ管理ノードで利用可能なツールは、ログに記録された書き込み、読み取り、削除操作の数と、これらの操作にかかった時間を要約します。

> その `audit-sum`このツールは主に、トラブルシューティング操作中にテクニカル サポートが使用することを目的としています。処理 `audit-sum`クエリは大量の CPU パワーを消費する可能性があり、 StorageGRID の操作に影響を与える可能性があります。

この例は、 `audit-sum`道具。この例では、プロトコル操作にかかった時間を示します。

```
  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
```

その `audit-sum`このツールは、監査ログ内の次の S3、Swift、および ILM 監査メッセージの数と時間を提供します。

> 機能が廃止されると、監査コードは製品およびドキュメントから削除されます。ここに記載されていない監査コードが発生した場合は、このトピックの以前のバージョンで古い SG リリースを確認してください。例： "StorageGRID 11.8 監査合計ツールの使用に関するドキュメント" 。

コード	説明	参照
アイデル	ILM による削除の開始: ILM がオブジェクトの削除プロセスを開始したときにログに記録します。	"IDEL: ILM による削除開始"
SDEL	S3 DELETE: オブジェクトまたはバケットを削除する成功したトランザクションをログに記録します。	"SDEL: S3 削除"
SGET	S3 GET: オブジェクトを取得したり、バケット内のオブジェクトを一覧表示したりするための成功したトランザクションをログに記録します。	"SGET: S3 ゲット"
シア	S3 HEAD: オブジェクトまたはバケットの存在を確認するために成功したトランザクションをログに記録します。	"シア：S3ヘッド"

コード	説明	参照
吐き出す	S3 PUT: 新しいオブジェクトまたはバケットを作成するための成功したトランザクションをログに記録します。	"スプット: S3 プット"
WDEL	Swift DELETE: オブジェクトまたはコンテナを削除する成功したトランザクションをログに記録します。	"WDEL: 迅速な削除"
WGET	Swift GET: オブジェクトを取得したり、コンテナー内のオブジェクトを一覧表示したりするための成功したトランザクションをログに記録します。	"WGET: Swift GET"
ウィー	Swift HEAD: オブジェクトまたはコンテナの存在を確認するために成功したトランザクションをログに記録します。	"WHEA: Swift HEAD"
WPUT	Swift PUT: 新しいオブジェクトまたはコンテナーを作成するための成功したトランザクションをログに記録します。	"WPUT: Swift PUT"

その `audit-sum`このツールは次のことができます。

- プレーンまたは圧縮された監査ログを処理します。例えば：

```
audit-sum audit.log

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

- 複数のファイルを同時に処理します。例えば：

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

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

- パイプからの入力を受け入れ、入力をフィルタリングしたり前処理したりすることができます。 `grep`命令またはその他の手段。例えば：

```
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、パイプ入力として圧縮ファイルを受け入れません。圧縮ファイルを処理するには、コマンドライン引数としてファイル名を指定するか、 `zcat`まずファイルを解凍するツールです。例えば：
>
> ```
> audit-sum audit.log.gz
>
> zcat audit.log.gz | audit-sum
> ```

コマンドライン オプションを使用すると、オブジェクトの操作とは別にバケットの操作を要約したり、バケット名、期間、またはターゲット タイプごとにメッセージの概要をグループ化したりできます。デフォルトでは、要約には最小、最大、平均操作時間が表示されますが、 `size (-s)`代わりにオブジェクトのサイズを確認するオプション。

使用 `help (-h)`利用可能なオプションを表示するには、オプションを選択します。例えば：

```
$ 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. 記載されているパスワードを入力してください `Passwords.txt`ファイル。
   c. ルートに切り替えるには、次のコマンドを入力します。 `su -`
   d. 記載されているパスワードを入力してください `Passwords.txt`ファイル。

      ルートとしてログインすると、プロンプトは `$`に `#`。

2. 書き込み、読み取り、ヘッド、削除操作に関連するすべてのメッセージを分析する場合は、次の手順に従います。
   a. 次のコマンドを入力します。 `/var/local/log/audit.log`分析するファイルの名前と場所を表します。

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

      この例は、 `audit-sum`道具。この例では、プロトコル操作にかかった時間を示します。

      ```
        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、SGET (S3 GET) 操作の平均時間が 1.13 秒で最も遅いですが、SGET 操作と SPUT (S3 PUT) 操作はどちらも最悪で約 1,770 秒という長い時間を示しています。

   b. 最も遅い10件の取得操作を表示するには、grepコマンドを使用してSGETメッセージのみを選択し、長い出力オプションを追加します。(`-l`) を使用してオブジェクト 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) とパスが含まれるため、監査ログで grep を実行して、これらの特定のオブジェクトに関連する他のメッセージを検索できます。

```
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、最も遅い 3 つの S3 GET リクエストは、サイズが約 5 GB のオブジェクトに対するものであり、他のオブジェクトよりもはるかに大きいことがわかります。サイズが大きいと、最悪の場合、取得時間が遅くなります。

3. グリッドに取り込まれるオブジェクトのサイズと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、サイズオプションを使用します。(`-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
```

この例では、SPUT の平均オブジェクト サイズは 2.5 MB 未満ですが、SGET の平均サイズははるかに大きくなります。 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取得されないことを示しています。

4. 昨日の取得が遅かったかどうかを判断したい場合:

   a. 適切な監査ログに対してコマンドを発行し、時間別グループオプションを使用します。(`-gt` ）の後に期間（例：15M、1H、10S）を続けます。

      ```
      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
```

これらの結果は、S3 GET トラフィックが 06:00 から 07:00 の間に急増したことを示しています。これらの時間では、最大時間と平均時間はどちらもかなり長くなっており、カウントが増加しても徐々に増加することはありません。これは、ネットワークまたはグリッドのリクエスト処理能力のどこかで容量が超過したことを示しています。

b. 昨日1時間ごとに取得されたオブジェクトのサイズを確認するには、サイズオプションを追加します。(`-s`) をコマンドに追加します:

```
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
```

これらの結果は、全体的な検索トラフィックが最大になったときに、非常に大規模な検索がいくつか発生したことを示しています。

c. さらに詳しく見るには、"監査説明ツール"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コマンドの出力が複数行になることが予想される場合は、 `less`監査ログ ファイルの内容を一度に 1 ページ (1 画面) ずつ表示するコマンド。

5. バケットに対する SPUT 操作がオブジェクトに対する SPUT 操作よりも遅いかどうかを確認するには、次の手順を実行します。

a. まずは `-go`オブジェクト操作とバケット操作のメッセージを個別にグループ化するオプション:

```
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
```

結果は、バケットに対する SPUT 操作は、オブジェクトに対する SPUT 操作とは異なるパフォーマンス特性を持つことを示しています。

b. SPUT操作が最も遅いバケットを特定するには、 `-gb`メッセージをバケットごとにグループ化するオプション:

```
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、 `-gb`そして `-s`オプション:

```
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
```

著作権に関する情報

商標に関する情報