Sunday, March 5, 2017

Auditing

Auditing

With MapR version 5.0 new auditing feature was added by MapR . The auditing features in MapR let you log audit records of cluster administration operations as well as operations on directories, files, and tables. Obvious use case is finding out who is doing what on your cluster.

Pre-req :-

Verify that the Audit feature is enabled, run this command:

```
[root@node9 ~]# maprcli config load  -json | grep mfs.feature.audit.support
   "mfs.feature.audit.support":"1",
```

If not below is command to turn the feature on.

maprcli config save -values {"mfs.feature.audit.support":"1"}

Note:- New installs from 5.x have it enabled and clusters upgraded to 5.x will have this feature disabled.

```
auditing enabled: "mfs.feature.audit.support":"1"
auditing disabled: "mfs.feature.audit.support":"0"
```

I - To enable or disable auditing of cluster-management operations on a MapR cluster, run

maprcli audit cluster -enabled true

To verify auditing is enabled run below command.

```
maprcli audit info  -json
{
 "timestamp":1488686193479,
 "timeofday":"2017-03-04 07:56:33.479 GMT-0800",
 "status":"OK",
 "total":1,
 "data":[
  {
   "data":{
    "enabled":"1",
    "maxSizeGB":"32",
    "retentionDays":"30"
   },
   "cluster":{
    "enabled":"1",
    "maxSizeGB":"NA",
    "retentionDays":"NA"
   }
  }
 ]
```

After enabling cluster level auditing, cluster management activities, including both MCS and MapR CLI commands, are written to the following log files:

```
/opt/mapr/logs/initaudit.log.json
/opt/mapr/logs/cldbaudit.log.json
/opt/mapr/mapr-cli-audit-log/audit.log.json
```

Example of logging for Maprcli command (under /opt/mapr/mapr-cli-audit-log)

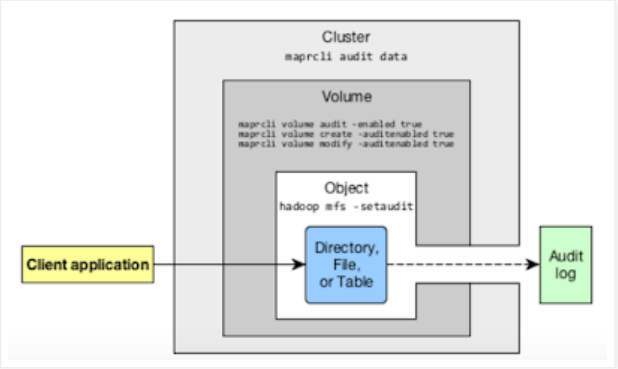
```
[root@node9 mapr-cli-audit-log]# maprcli node cldbmaster
cldbmaster
ServerID: 5979527104878929364 HostName: node9.maprlab.local
[root@node9 mapr-cli-audit-log]# pwd
/opt/mapr/mapr-cli-audit-log
[root@node9 mapr-cli-audit-log]# tail -1 audit.log.json
{"timestamp":{"$date":"2017-03-05T04:10:37.593Z"},"uid":0,"ipAddress":"10.10.70.109","command":"node cldbmaster","arguments":{},"status":0}
```

Example of logging for CLDB and authorization (Under /opt/mapr/logs)

```
[root@node9 logs]# tail -1 cldbaudit.log.json
{"timestamp":{"$date":"2017-03-05T04:00:47.720Z"},"resource":"cluster","operation":"fileServerLookup","uid":2000,"clientip":"10.10.70.109","status":0}
[root@node9 logs]# tail -5 authaudit.log.json
{"timestamp":{"$date":"2017-03-04T21:03:47.687Z"},"resource":"cluster","operation":"passwordAuth","username":"mapr","clientip":"172.30.1.78","status":200}
```

II - To enable or disable auditing of data-access operations:

Auditing of filesystem, table, and stream operations needs to be enabled at three separate levels (the cluster level, through the maprcli audit datacommand; the volume level, through any of the three volume commands shown in the diagram; and the level of the individual directory, file, table, or stream.) . If auditing is not enabled at any one of these levels, operations on an object are not logged.



1) Enabling data auditing at the cluster level (item 1). Data auditing parameter is like master switch for enabling or disabling audit logging at the data level for the complete cluster, This command does not cause auditing to start for operations within those volumes. It only sets a flag that says you allow auditing of individual volumes to be enabled with the `maprcli volume audit` command.

maprcli audit data -enabled true -maxsize 50 -retention 99

Maxsize - Max size of audit logs retention - Days to retain audit logs

Verify command updated the values and enabled data auditing .

```
[root@node9 mapr-cli-audit-log]# maprcli audit info -json
{
 "timestamp":1488690406118,
 "timeofday":"2017-03-04 09:06:46.118 GMT-0800",
 "status":"OK",
 "total":1,
 "data":[
  {
   "data":{
    "enabled":"1",
    "maxSizeGB":"50",
    "retentionDays":"99"
   },
   "cluster":{
    "enabled":"1",
    "maxSizeGB":"NA",
    "retentionDays":"NA"
   }
  }
 ]
}
```

2) To enable or disable auditing for a particular volume, we run the `maprcli volume audit` command.

maprcli volume audit -name mapr.opt -enabled true -coalesce 3

coalesceInterval - Interval of time during operations on one file from same client IP address are logged only once.

To verify that auditing is enabled for mapr.opt volume.

```
[root@node9 mapr-cli-audit-log]# maprcli volume info -name mapr.opt -json | grep -i 'audited\|coalesce'
   "audited":1,
   "coalesceInterval":3,
```

3) With data auditing enabled at both the cluster and volume level, it's now time to enable auditing at specific directories, files, tables or stream. This be done using the hadoop mfs setaudit command as shown below:

```
hadoop mfs -setaudit on|off <dir|file|table>

example set audit on dir :

hadoop mfs -setaudit on /opt/abizer
```

After enabling auditing, please verify the audit bit has been set on your directory:

```
hadoop mfs -ls /opt/



Found 1 items
```

```
drwxr-xr-x  Z U A   - root root          0 2017-03-04 13:07  268435456 /opt/abizer


      p 2066.32.131274  node9.maprlab.local:5660
```

Here :
drwxr-xr-x Z U U < auditing disabled, last bit being 'U'
drwxr-xr-x Z U A < auditing enabled, last bit being 'A'

Note : Enabling auditing on a directory does not enable auditing on the files that already exist in the directory, though new files and directories created in the directory will have auditing enabled. Wild cards * are not supported.

With this we are all set and every activity on /opt/abizer dir is being audited and logged to the MapR file system at /var/mapr/local/node9.maprlab.local/audit/5660.

Example.
hadoop fs -ls /var/mapr/local/node9.maprlab.local/audit/5660
Found 20 items

```
-rw-------   2 mapr mapr          0 2017-02-21 13:57 /var/mapr/local/node9.maprlab.local/audit/5660/DBAudit.log-2017-02-21-002.json
-rw-------   2 mapr mapr          0 2017-02-09 10:13 /var/mapr/local/node9.maprlab.local/audit/5660/ExpandAudit.log-2017-02-21-002.json
-rw-------   2 mapr mapr          0 2017-02-09 10:13 /var/mapr/local/node9.maprlab.local/audit/5660/FSAudit.log-2017-02-21-002.json
-rw-------   2 mapr mapr          0 2017-02-09 10:13 /var/mapr/local/node9.maprlab.local/audit/5660/Vollist_DBAudit.log-2017-02-21-002
-rw-------   2 mapr mapr          0 2017-02-21 13:57 /var/mapr/local/node9.maprlab.local/audit/5660/Vollist_FSAudit.log-2017-02-21-002
```

Example logging file creation and deletion :-

```
[root@node9 ~]# id abizer
uid=5003(abizer) gid=5003(abizer) groups=5003(abizer)
[abizer@node9 root]$ hadoop fs -touchz /opt/abizer/testing123

[abizer@node9 root]$ hadoop fs -rm  /opt/abizer/testing123
17/03/05 02:52:17 INFO Configuration.deprecation: io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
17/03/05 02:52:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /opt/abizer/testing123

[root@node9 ~]# hadoop fs -cat /var/mapr/local/node9.maprlab.local/audit/5660/FSAudit.log-2017-03-05-001.json | grep testing123
CREATE log :
{"timestamp":{"$date":"2017-03-
05T10:51:36.256Z"},"operation":"CREATE","uid":5003,"ipAddress":"10.10.70.109","parentFid":"2066.32.131274","childFid":"2066.45.131300","childName":"testing123","volumeId":99999763,"status":0}

DELETE log :
{"timestamp":{"$date":"2017-03-
05T10:52:17.595Z"},"operation":"LOOKUP","uid":5003,"ipAddress":"10.10.70.109","srcFid":"2066.32.131274","dstFid":"2066.45.131300","srcName":"testing123","volumeId":99999763,"status":0}
{"timestamp":{"$date":"2017-03-
05T10:52:17.619Z"},"operation":"DELETE","uid":5003,"ipAddress":"10.10.70.109","parentFid":"2066.32.131274","childFid":"2066.45.131300","childName":"testing123","volumeId":99999763,"status":0}
```

Example logging Table creation :-

```
[abizer@node9 root]$ maprcli table create -path /opt/abizer/testingTable123
[abizer@node9 root]$ hadoop fs -ls /opt/abizer/testingTable123
tr--------   - abizer abizer          2 2017-03-05 03:01 /opt/abizer/testingTable123
```

CREATE log :

```
 hadoop fs -cat /var/mapr/local/node9.maprlab.local/audit/5660/DBAudit.log-2017-03-05-001.json
{"timestamp":{"$date":"2017-03-
05T11:01:16.482Z"},"operation":"DB_TABLECREATE","uid":5003,"ipAddress":"10.10.70.109","srcName":"testingTable123","volumeId":99999763,"parentFid":"2066.32.131274","tableFid":"2066.46.131302
","status":0}
```

Utility to review logs easily : expandaudit

As operations are performed on the directories, files, tables and streams that you are auditing, the audit logs capture records of those operations. Those records identify the affected directories, files, and tables by means of file IDs, the volumes on which the operations took place by means of volume id, and the users who performed the operations by means of user IDs. These IDs are used instead of names in the audit records because fetching the actual names of these objects and users in real-time is costly in terms of performance. Now we can use the expandaudit utility, to create copies of logs files in which the IDs are resolved into names and inserted into the audit records for readability, volume id is resolved into volume names and inserted into the audit records for readability lastly also FID/Tablet are resolved in Filename/Table and inserted into the audit records for readability. Also since this logs are for specific volume it only pulls data of interest i.e audit logs for volume under investigation.

```
[root@node9 tmp]# /opt/mapr/bin/expandaudit -volumename mapr.opt -o /opt/
ExpandAuditLogs with args:
volumeName = mapr.opt
Output Directory for expanded audit logs = /opt/
Input Directory for audit logs = /var/mapr/local/
HostName = node9.maprlab.local
Please look in mapr logs location for errlog
Using CLDB Extracted VolumeId = 99999763
Processing audit logs from Node = node9.maprlab.local
audit logs path = maprfs:///var/mapr/local/node9.maprlab.local/audit
Processing audit logs from Node = node9.maprlab.local
audit logs path = maprfs:///var/mapr/local/node9.maprlab.local/audit/5660
Phase1 Audit Log Expansion Done. Starting Phase2
Phase2 Audit Log Expansion Done. Exiting
```

Example of Audit table log which was collected earlier but now in more readable format:

```
[root@node9 tmp]# hadoop fs -cat /opt/99999763/node9.maprlab.local/5660/DBAudit.log-2017-03-05-001.part.json | grep testingTable123
```

{"timestamp":{"$date":"2017-03-05T11:01:16.482Z"},"operation":"DB_TABLECREATE","user":"abizer","uid":5003,"ipAddress":"10.10.70.109","srcName":"testingTable123","VolumeName":"mapr.opt","volumeId":99999763,"parentPath":"/opt/abizer","parentFid":"2066.32.131274","tablePath":"/opt/abizer/testingTable123","tableFid":"2066.46.131302","status":0}

No comments:

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)