Technical Report

Best-practice guidelines for ONTAP File System Analytics

Solution deployment

Karthikeyan Nagalingam, NetApp
November 2020 | TR-4867-DEPLOY

Abstract

This document describes the NetApp® ONTAP® File System Analytics architecture, integration with ONTAP System Manager, REST API communication for external applications, challenges in analytics, and how ONTAP File System Analytics provides solutions for various problems you might face.

TABLE OF CONTENTS

Introduction **4**

ONTAP File System Analytics **4**

Technology overview 4

Architecture 5

ONTAP File System Analytics integration with System Manager **5**

Enable File System Analytics 6

System Manager Views for File System Analytics 6

ONTAP File System Analytics Initialization Scan **9**

Number of Directories and Files 9

FlexVols and FlexGroups 10

Scanning Multiple Volumes 10

Client Access 10

Remote Hardlinks 11

Granular Data 12

Hardware Platform 12

ONTAP File System Analytics Rest API **12**

System Manager integration with ONTAP File System Analytics 12

Application integration with ONTAP File System Analytics 13

Challenges **17**

Use cases **18**

ONTAP File System Analytics - Capabilities **19**

Least and most-changed workloads 19

Largest folders 23

Locate large numbers of files 25

Activity behavior 26

JSON format of analytics REST API 26

Conclusion **27**

Appendix **27**

Where to find additional information **30**

Version history **30**

LIST OF TABLES

Table 1) Expected scan duration for datasets. 9
Table 2) REST API details....14
Table 3) Analytics REST API JSON. 26

LIST OF FIGURES

Figure 1) ONTAP File System Analytics: high-level data. 4
Figure 2) ONTAP File System Analytics architecture. 5
Figure 3) System Manager List and Table View for ONTAP File System Analytics. 7
Figure 4) System Manager Graphical View, ONTAP File System Analytics. 8
Figure 5) System Manager Table View for ONTAP File System Analytics. 8
Figure 6) System Manager Summery Panel for ONTAP File System Analytics. 9
Figure 7) FlexGroups automatically distribute load. 10
Figure 8) Pause the scan from System Manager 11
Figure 9) Resume the scan from System Manager 11
Figure 10) The scanner must start new tasks for remote files. 12
Figure 11) System Manager integration with ONTAP File System Analytics and API reference. 13
Figure 12) Sample flow of ONTAP File System Analytics API documentation for the analytics API. 14
Figure 13) ONTAP File System Analytics - customer challenges. 17
Figure 14) ONTAP File System Analytics major use cases. 18
Figure 15) System Manager view for most and least changed workloads. 20
Figure 16) System Manager view for the largest folders. 24
Figure 17) System Manager aberrant activity detection. 26

Introduction

This document describes the NetApp® ONTAP® File System Analytics architecture, integration with ONTAP System Manager, REST API communication for external applications, challenges in analytics, and how ONTAP File System Analytics provides solutions for various problems you might face.

ONTAP File System Analytics

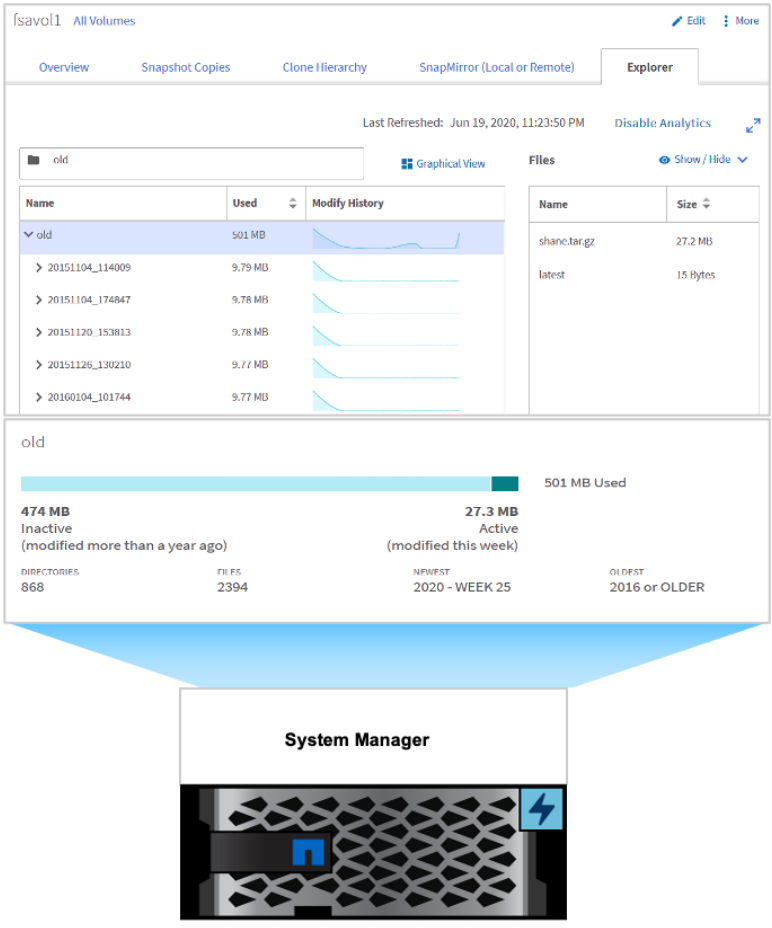
ONTAP File System Analytics, a new feature introduced in ONTAP 9.8, is a framework for collecting and displaying data about NetApp FlexGroup and NetApp® FlexVol® volume contents, providing visibility into capacity access and usage for files and directories over time without an external tool. This real-time visibility helps you with effective data management and operation, such as quality-of-service changes to throughput and moving volumes to primary storage and secondary storage.

Technology overview

ONTAP File System Analytics collects and aggregates data in real time, providing detailed information concerning file and directory count, file-age histograms, capacity use, and other parameters.

We can visualize data using industry standard measures such as hot, warm, and cold. File System Analytics provides similar data in using active, normal, and inactive labels.

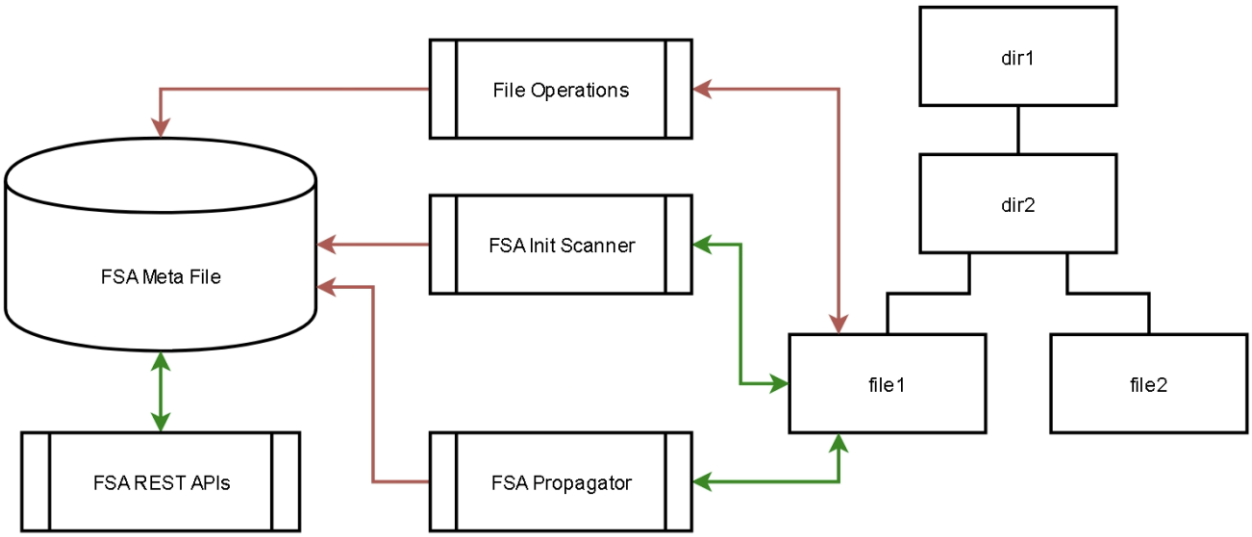
Figure 1) ONTAP File System Analytics: high-level data.



Architecture

The ONTAP File System Analytics architecture has five components: the Init Scanner, the Propagator, the Meta File, File Operations, and REST APIs (Figure 2; red arrows indicate read-and-write operations, whereas green arrows indicate read-only operations).

Figure 2) ONTAP File System Analytics architecture.



When a privileged user such as the admin enables analytics in System Manager, from the CLI, or with a REST API, the ONTAP filesystem Init Scanner runs in parallel on multiple directories in a FlexVol volume. It also runs on multiple constituents in a FlexGroup volume at the same time and passes the collected details to the Meta file. The runtime of the scanner is proportional to number of inodes. Each FlexVol volume has one Meta File and each constituent has one Meta File in a FlexGroup. The Meta File stores the details in the key-value format like a database; the key is the identity of a directory and value is the File System Analytics data.

The File System Analytics Propagator propagates the accumulated File System Analytics changes at each level up the directory hierarchy. It runs as a process in the background to provide near real-time aggregation for both FlexGroup and FlexVol volumes. File operations record filesystem changes and report to the Meta File. Its interactions are in-line and lightweight to minimize performance effects.

REST APIs can query a directory to collect the File System Analytics record from the Meta File without scanning the filesystem. By default, the results are in the JSON format and you can perform sorting and filtering and set the timeout and limits for the returning records. You can also turn on/off the ONTAP File System Analytics using ONTAP REST APIs.

ONTAP File System Analytics integration with System Manager

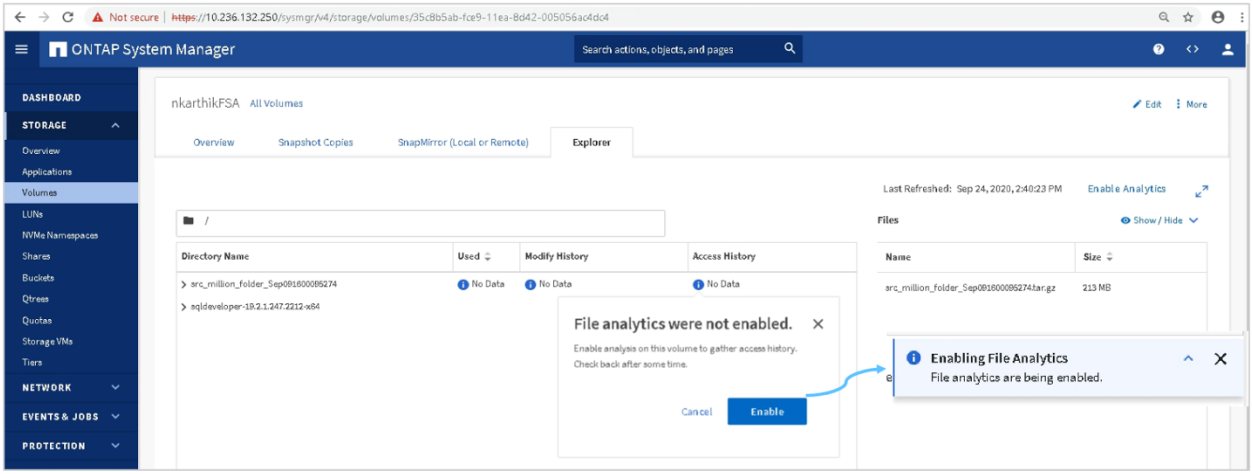
ONTAP File System Analytics can be displayed using ONTAP System Manager. Normally, System Manager is operated by privileged users such as the admin. This user must have read-only access to ONTAP API calls that can collect details from File System Analytics through System Manager from any directory level.

Enable File System Analytics

You can enable File System Analytics to collect and display usage data using System Manager, the ONTAP CLI, or REST APIs. You can enable File System Analytics when you create a new volume or upgrade a system with volumes for ONTAP 9.8 or later.

Perform the following tasks to enable File System Analytics:

1. Navigate to Storage > Volumes and select the desired volume.
2. Navigate to Explorer and either enable or disable analytics.



Depending on the size and content of the volume, enabling analytics might take some time while ONTAP processes data in the volume. You can view the progress from System Manager or the ONTAP CLI by using the `volume analytics show` command.

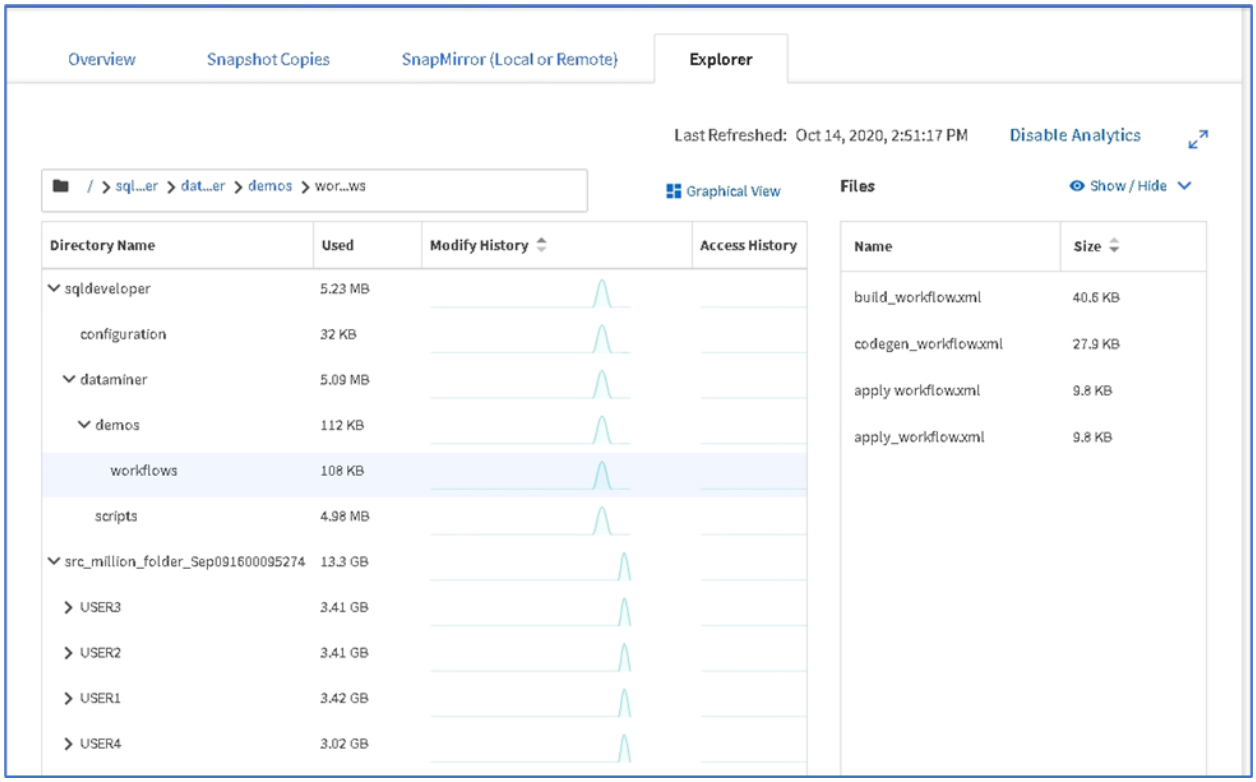
System Manager Views for File System Analytics

System Manager provides four views for File System Analytics: list view, graphical view, files view, and summary view.

List view

- Tree view of folders and subfolders
- Expandable lists in sorted order for showing the name, used size, modify history, and access history.

Figure 3) System Manager List and Table View for ONTAP File System Analytics.

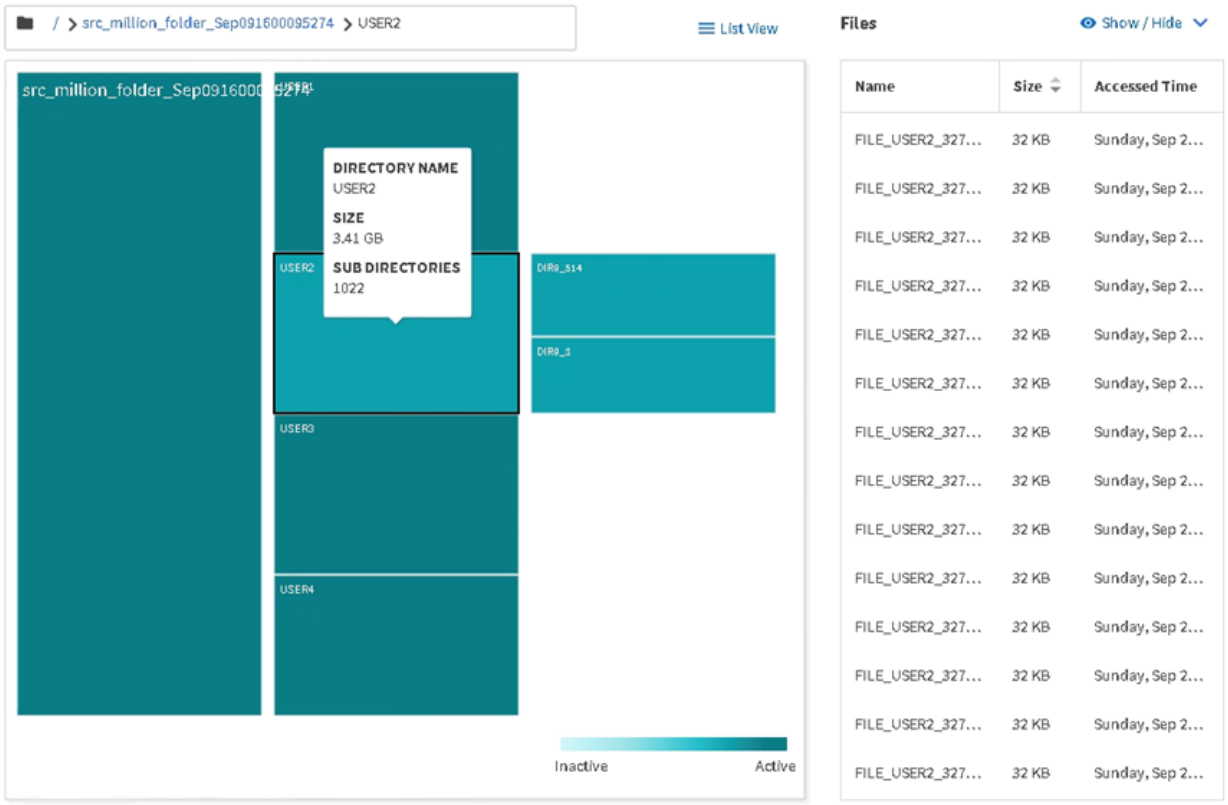


Note: In the current release, the default Sorting parameter is disabled for any folder having more than 3000 folders at the same level or in a single folder. However, you can use an API call without limit.

Graphical view

- The box size of the directory is proportional to the size of the directories.
- When you move the cursor on the top of the folder, you can view the name, size, number of sub-directories in that folder.
- In the first release, the top three folders are displayed.

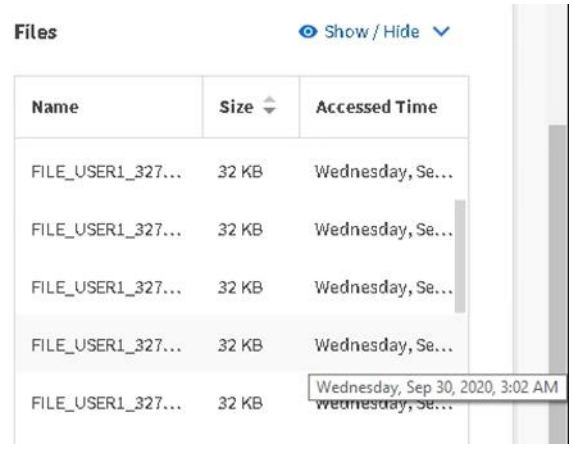
Figure 4) System Manager Graphical View, ONTAP File System Analytics.



Files view

- The file view provides the file name, size, and accessed time for the selected object in the directory list.
- Select the file to view the access time and date.

Figure 5) System Manager Table View for ONTAP File System Analytics.

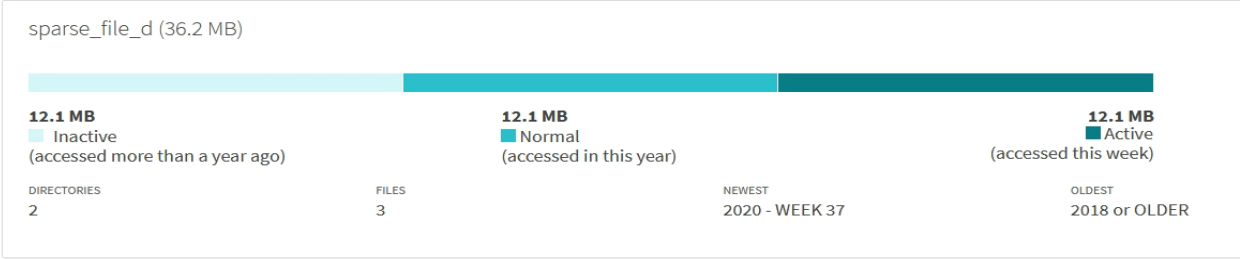


Note: There is not a dynamic refresh for the files view. Therefore, when you add a new file to the folder, you can select another tab and return to see the updated information.

Summary panel

- Provides the number of directories and files, and it also present age information.
- Presents data in with the labels active, normal, and inactive.
  - Active data are accessed less than one week.
  - Inactive data are accessed more than one year.
  - Normal data are accessed between one week to one year.

Figure 6) System Manager Summery Panel for ONTAP File System Analytics.



Note: ONTAP File System Analytics is not compatible with NetApp FlexCache® volumes and SAN volumes. In this release, it only works with NAS volumes.

ONTAP File System Analytics Initialization Scan

The scan is a required step for FSA to gather information about existing files in the file system. The information is all metadata for the files that is required for FSA to function such as size, access time, and modify time. After the scan completes, FSA is continuously updated in real time as the filesystem changes without the need to run the scan again.

When analytics is enabled, the scan is automatically started. It traverses directories in parallel collecting analytics data to be processed and then stored in the analytics meta file. The parallelism is achieved by running tasks for each directory. Each task will scan the files in the directory, and queue new tasks for the scanner.

Number of Directories and Files

The scan's duration is directly proportional to the number of directories and files on the volume. The scan does not read any data contained in the files. Only metadata required for analytics such as file size are gathered. Therefore, the actual size of files does not affect the duration of the scanner.

While both directories and files will contribute to the total scan time, recursive traversal of directories is more intensive than reading file metadata. In most cases, the number of directories in the filesystem will heavily affect the scan duration. A volume containing 4.5 million directories and 22 million files will take up to 45 minutes. The time can increase up to 6 hours under scenarios described in the sections below.

Table 1) Estimated scan duration for datasets.

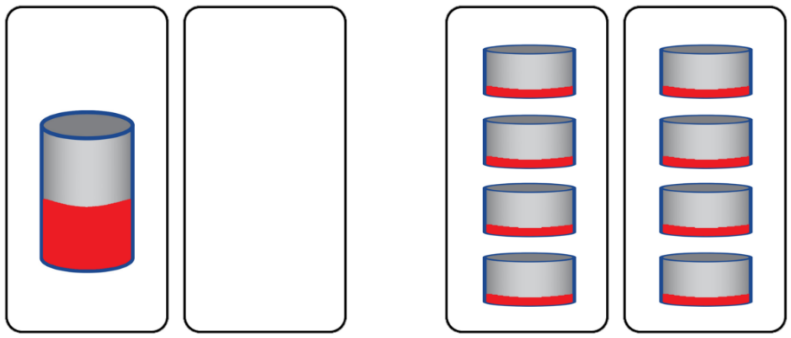
Directories	Files	Minimum Time	Additional Time
0 **to** 4.2 million	0 **to** 20 million	up to 40 minutes	up to 5.5 hours
0 to 4.5 million	0 to 22 million	up to 45 minutes	up to 6 hours
0 to 17 million	0 to 340 million	up to 4.5 hours	up to 4 days

Note: Estimates are provided as a general guideline for ONTAP 9.13. Scan durations may fall outside the range of the provided estimates.

FlexVols and FlexGroups

The scanner will scan directories in parallel on each FlexVol. This is also true for FlexGroups as the scanner will also run on all constituents in parallel. Internal limits dictate how many directories are scanned in parallel on each FlexVol or FlexGroup volume. Both volume types are subject to the same limit, so the volume type is not a limiting factor if the volumes reach the speed limit. What will affect the scanner speed is when the cluster has to use resources to perform other tasks instead of the scan. This is how the FlexGroups differ from FlexVols. FlexGroups have the ability to distribute load across the cluster, including the scanner load. Using the entire cluster's resources more efficiently lets the volume perform other tasks while the scan is running such as serving client traffic.

Figure 7) FlexGroups automatically distribute load.



Scanning Multiple Volumes

The scanner will generate load on a volume and consume system resources. The more volumes that are being scanned, the more load is generated. Scanner limits are only enforced at a per volume level and starting too many scans that share resources can slow down the scans.

It is recommended to start the scanner on volumes that do not share aggregates. You can see which aggregates are currently hosting which volumes. FlexGroups will report having more than one aggregate if it was deployed following NetApp recommended guidelines.

```
::> volume show -volume flexvol,flexgroup -fields aggr-list
vserver volume    aggr-list
------- --------- ----------------------
vs0     flexgroup aggr1,aggr2,aggr3,aggr4
vs0     flexvol   aggr1
2 entries were displayed.
```

Client Access

Volumes can continue to serve client traffic while the scan is running. As client traffic also generates load and consumes system resources, it will cause the scan to take longer.

If serving client data is critical, the FSA scan should be run during periods of lower client traffic. This will ensure that the cluster dedicates more of its resources towards serving data instead of the scan. Starting from the 9.12.1 release, the scan can be paused during periods of high traffic, and then resumed later from exactly where it left off.

All files created, deleted, or modified while the scan is paused will be included in FSA. If the file changes occur in a directory that has not been scanned, the directory will be included in FSA data after the scan is resumed.

```
::> volume analytics initialization pause -vserver vs0 -volume vol1

::> volume analytics initialization resume -vserver vs0 -volume vol1
```

Figure 8) Pause the scan from System Manager

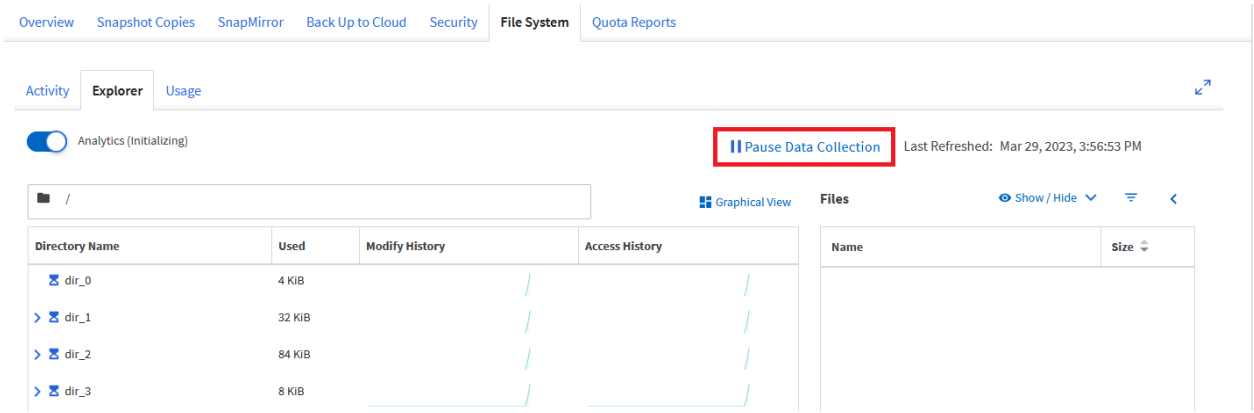
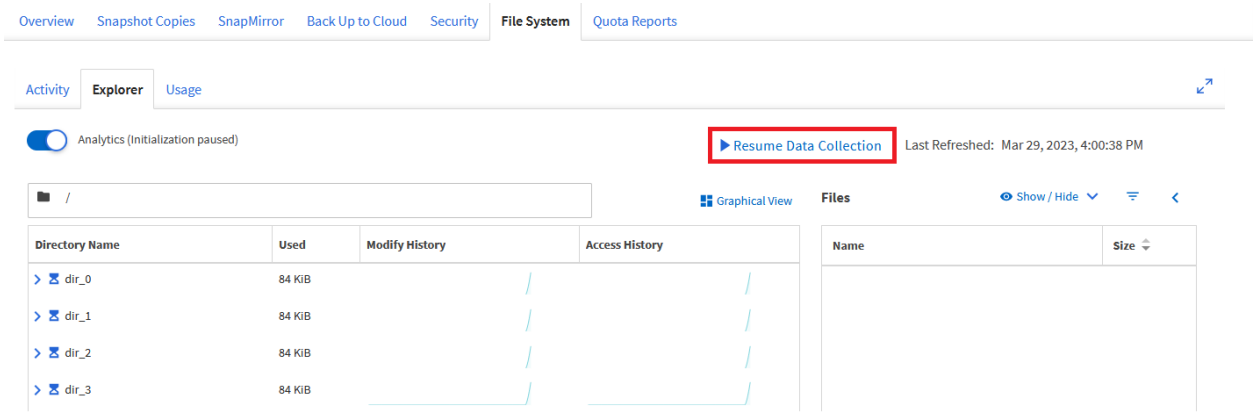


Figure 9) Resume the scan from System Manager



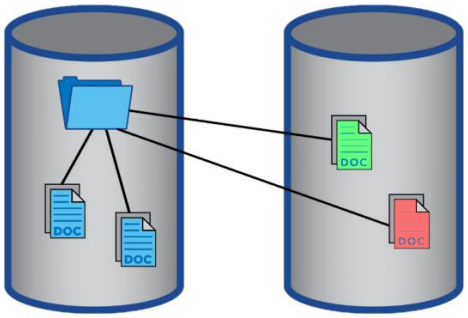
Remote Hardlinks

Remote hardlinks are what allows a FlexGroup volume to distribute data between its constituents. Both directories and files can employ remote hardlinks to be stored on a different constituent than the parent directory. Scanning a file that is on the same constituent as its parent directory is normally a fast operation. However, for remote hardlinks, a task must be started on the constituent volume that hosts the file or directory. The scanner penalty for remote directories is smaller than local directories as the scanner will start a new task for a new directory regardless. However for files, a remote file will incur the task penalty that a local file on the same constituent will not.

The FlexGroup will determine whether or not to create a remote hardlink at the time of the file creation. A file can also become a remote hardlink during a rename when the parent directory is on a different constituent. When there is a low amount of free space on a FlexGroup, there may be an increase in the number of remote hardlinks that are created. FlexGroups that operate in low space conditions for

extended periods of time may have a disproportionate number of remote hardlinks that will increase the scan time.

Figure 10) The scanner must start new tasks for remote files.



Granular Data

ONTAP 9.12.1 introduced the Granular Data management feature for FlexGroups. This feature allows files on a FlexGroup to be moved non-disruptively from one constituent to another.

Granular Data is primarily used by FlexGroup rebalancing. Files that are moved during the rebalancing process will gain an additional management layer just as with remote hardlinks. A file can have both the remote hardlink and Granular Data layers. The rebalancing feature has controls that can limit what kind of files will be chosen for rebalancing. Depending on the settings used for rebalancing, there can be many files that gain the additional layer. Like with remote hardlinks, a file that has this layer will increase the scan time.

Hardware Platform

As with the comparison with FlexVols and FlexGroups, a faster hardware platform does not necessarily translate into an increase in the scan speed. All hardware platforms are subject to the same internal volume limits described in the FlexVols and FlexGroups section. Higher end hardware platforms have more overall resources such that the FSA scan will use a smaller percentage of the total. The scan is less likely to slow down due to other cluster traffic on higher end hardware platforms.

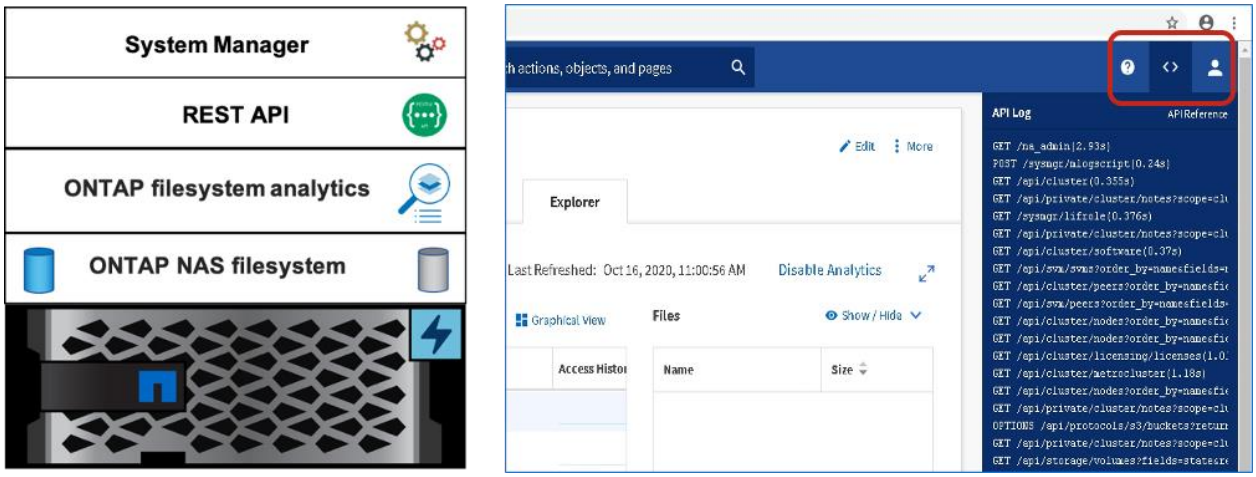
ONTAP File System Analytics Rest API

REST APIs dominate service application development. Currently, users require REST API support for a product for their own development.

System Manager integration with ONTAP File System Analytics

NetApp System Manager collects data from ONTAP File System Analytics using REST APIs to build list, graphical, tree, and summary views. System Manager automates the entire workflow for REST API communication.

Figure 11) System Manager integration with ONTAP File System Analytics and API reference.



For quick reference, System Manager provides API reference information on the Explore tab and the <> icon in the top right corner. This reference allows users to review the APIs used by System Manager for each operation.

Note: ONTAP File System Analytics does not have role-based access control (RBAC), but you can block users from using the CLI by disabling analytics APIs for the user.

Application integration with ONTAP File System Analytics

A customer will commonly want to integrate ONTAP File System Analytics with applications such as Grafana. ONTAP File System Analytics supports REST APIs, and you can create your own views that are not in System Manager.

See the following example to get folder and file counts in sorted order (descending). In the following example, the variables are user, pw, and ontap_cluster_management_ip with values that are already provided.

```
root@scspr1882834014 demo]# curl -siku $user:$pw  --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/1f911d57-e279-11ea-90fc-
005056a7adc6/files?fields=analytics.file_count&order_by=analytics.file_count+desc" | egrep
'name|file_count'
      "name": ".",
        "file_count": 735
      "name": "logs",
        "file_count": 628
      "name": "www",
        "file_count": 76
      "name": "misc",
        "file_count": 26
      "name": ".."
      "name": ".snapshot"
      "href": "/api/storage/volumes/1f911d57-e279-11ea-90fc-
005056a7adc6/files?fields=analytics.file_count&order_by=analytics.file_count+desc"
```

To see the ascending order in the above example, change the keyword `desc` to `asc`.

You can communicate with ONTAP APIs, and specifically analytics APIs, using the ONTAP cluster management IP API documentation from the browser. See the following example, which provides analytics information from the root folder of the volume:

Figure 12) Sample flow of ONTAP File System Analytics API documentation for the analytics API.

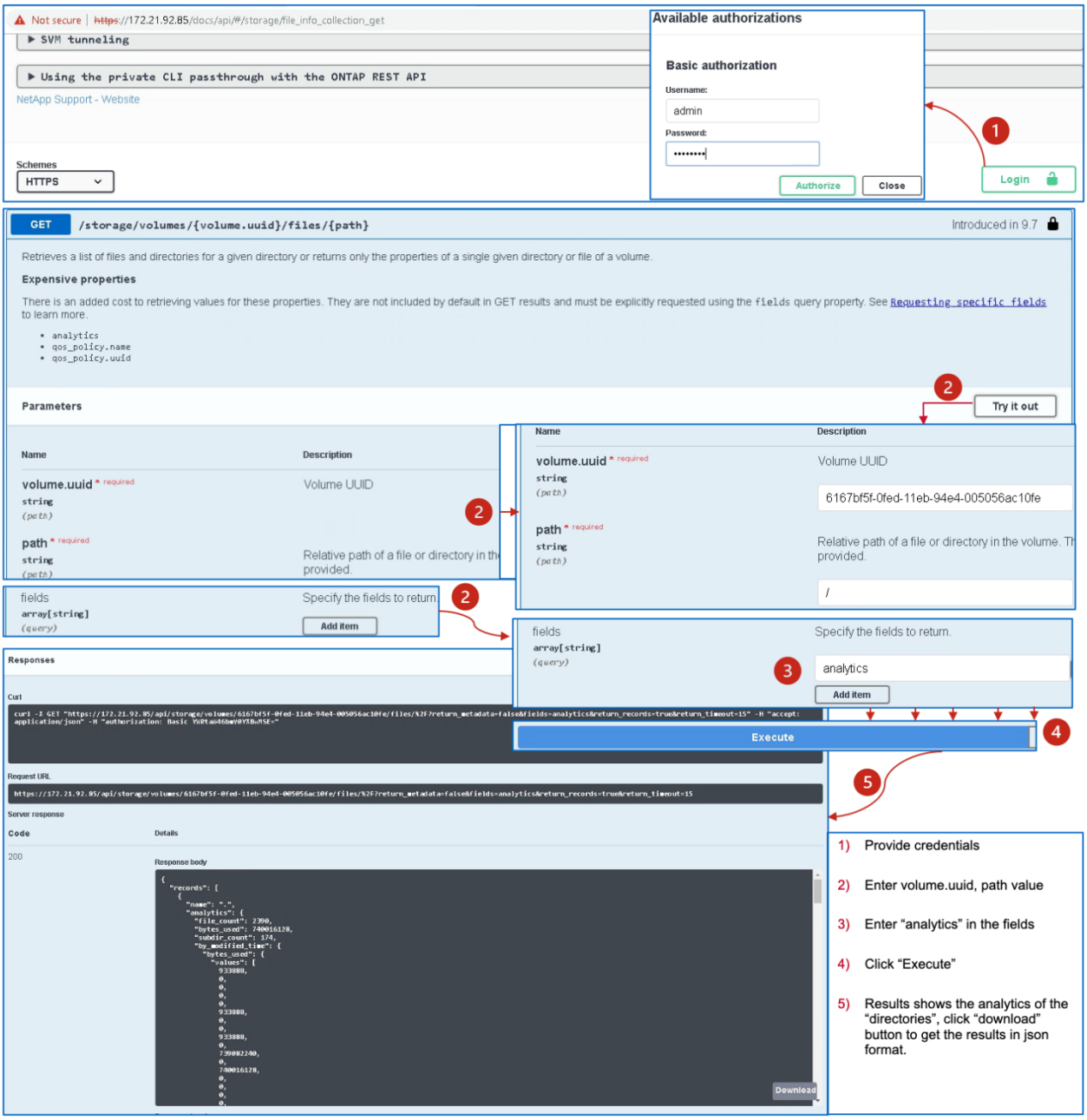


Table 2 provides details about the analytics REST APIs.

Table 2) REST API details.

Analytics REST API	Type	Meaning
`analytics.by_modified_time.bytes_used.oldest_label`	String	Filter by `analytics.by_modified_time.bytes_used.oldest_label`
`analytics.by_modified_time.bytes_used.percentages`	Number	Filter by `analytics.by_modified_time.bytes_used.percentages`

Analytics REST API	Type	Meaning
analytics.by_modified_time.bytes_used.values	Integer	Filter by analytics.by_modified_time.bytes_used.values
analytics.by_modified_time.bytes_used.newest_label	String	Filter by analytics.by_modified_time.bytes_used.newest_label
analytics.by_modified_time.bytes_used.labels	String	Filter by analytics.by_modified_time.bytes_used.labels
analytics.file_count	Integer	Filter by analytics.file_count
analytics.bytes_used	Integer	Filter by analytics.bytes_used
analytics.subdir_count	Integer	Filter by analytics.subdir_count
analytics.by_accessed_time.bytes_used.oldest_label	String	Filter by analytics.by_accessed_time.bytes_used.oldest_label
analytics.by_accessed_time.bytes_used.percentages	Number	Filter by analytics.by_accessed_time.bytes_used.percentages
analytics.by_accessed_time.bytes_used.values	Integer	Filter by analytics.by_accessed_time.bytes_used.values
analytics.by_accessed_time.bytes_used.newest_label	String	Filter by analytics.by_accessed_time.bytes_used.newest_label
analytics.by_accessed_time.bytes_used.labels	String	Filter by analytics.by_accessed_time.bytes_used.labels

The analytics.by_access_time.bytes_used and analytics.by_modify_time.bytes_user APIs has three items such as labels,newest_label and oldest_label.Lablels is a string indicating the period of time the corresponding data is associated with. Newest_label (most recent label) and Oldest_label (oldest label) with the non-zero value in the accessed and modified histogram value.

See the following sample analytics.file_count API results in descending order (based on the template script provided in the appendix).

```
[root@scspr1936701025 ~]# bash apitemplate.sh
User : admin
Password :
Server : 10.236.153.165
Volume UUID : 2538a931-13c1-11eb-bfde-005056a772a7
choose the API from the list if not enter the API name:
 1. analytics.file_count
 2. analytics.by_modified_time.bytes_used.oldest_label
 3. analytics.by_modified_time.bytes_used.percentages
 4. analytics.by_modified_time.bytes_used.values
 5. analytics.by_modified_time.bytes_used.oldest_label
 6. analytics.by_modified_time.bytes_used.newest_label
 7. analytics.by_modified_time.bytes_used.labels
 8. analytics.bytes_used
 9. analytics.subdir_count
 10. analytics.by_accessed_time.bytes_used.oldest_label
```

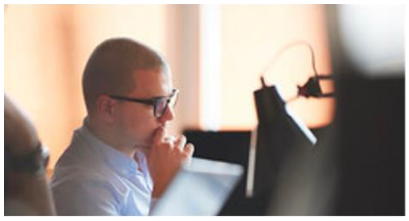
```
 11. analytics.by_accessed_time.bytes_used.percentages
 12. analytics.by_accessed_time.bytes_used.values
 13. analytics.by_accessed_time.bytes_used.newest_label
 14. analytics.by_accessed_time.bytes_used.labels
 0. Others
choose the number or '0' for provide API name : 1
API choosen : analytics.file_count
Full path. default(/): /
Do you want to call the API with default options. (y/n): default(y):n
order_by filedname. default is API name(analytics.file_count):
order_by. asc/desc : default (asc): Desc
Number of seconds to allow the call to execute before returning: default (15): 10
user pw server volumeuuid apiname path fieldname order_by return_records max_records
return_timeout
admin xxxx 10.236.153.165 2538a931-13c1-11eb-bfde-005056a772a7 analytics.file_count %2F
analytics.file_count desc 10
curl -siku admin:xxx  --request GET "https://10.236.153.165/api/storage/volumes/2538a931-13c1-
11eb-bfde-
005056a772a7/files/%2F?fields=analytics.file_count&order_by=analytics.file_count+desc&return_time
out=10"
HTTP/1.1 200 OK
Date: Mon, 26 Oct 2020 21:18:04 GMT
Server: libzapid-httpd
X-Content-Type-Options: nosniff
Cache-Control: no-cache,no-store,must-revalidate
Content-Type: application/hal+json
Transfer-Encoding: chunked

{
  "records": [
    {
      "name": ".",
      "analytics": {
        "file_count": 137355
      }
    },
    {
      "name": "src_million_folder_Sep091600095274",
      "analytics": {
        "file_count": 134861
      }
    },
    {
      "name": "sqldeveloper-19.2.1.247.2212-x64",
      "analytics": {
        "file_count": 2390
      }
    },
    {
      "name": "20190626_162237",
      "analytics": {
        "file_count": 31
      }
    },
    {
      "name": "20190626_161806",
      "analytics": {
        "file_count": 24
      }
    },
    {
      "name": "20190626_162128",
      "analytics": {
        "file_count": 24
      }
    },
    {
      "name": "20170504_083851",
      "analytics": {
        "file_count": 8
      }
```

```
    },
    {
      "name": "20170504_090829",
      "analytics": {
        "file_count": 8
      }
    },
    {
      "name": "20170504_090917",
      "analytics": {
        "file_count": 8
      }
    },
    {
      "name": ".."
    },
    {
      "name": "src_million_folder_Sep091600095274.tar.gz"
    },
    {
      "name": ".snapshot"
    }
  ],
  "num_records": 12,
  "_links": {
    "self": {
      "href": "/api/storage/volumes/2538a931-13c1-11eb-bfde-
005056a772a7/files/%2F?fields=analytics.file_count&order_by=analytics.file_count+desc&return_time
out=10"
    }
  }
}[root@scspr1936701025 ~]#
```

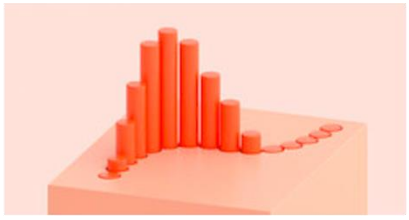

Challenges

Customers have the following top three challenges when working with file analytics of NAS data.

Figure 13) ONTAP File System Analytics - customer challenges.



Need actionable intelligent insights



Real-time analytics

Hierarchical with granularity

Need actionable intelligent insights

Customers wants to derive insights from their files and folders; for example, they might want to know where most of the changes happened based on the modification time or the access time. What are the largest folders? Is the data hot? Such information is needed so that you can take necessary actions such as moving hot data to primary, high-speed storage and cold data to secondary or cloud storage with proper QoS controls.

Real-time analytics

Customers need their file-analytics report online in real-time to provide their stakeholders with critical information. Reports containing system history present previous data behavior and provide additional tools for data management. These capabilities are particularly helpful for sensitive data.

Hierarchical with granularity

Customers wants to see their users home folders and drill down further based on size, number of files, modification time and access time of the data. It can be difficult for larger organizations to evaluate the behavior of home folders.

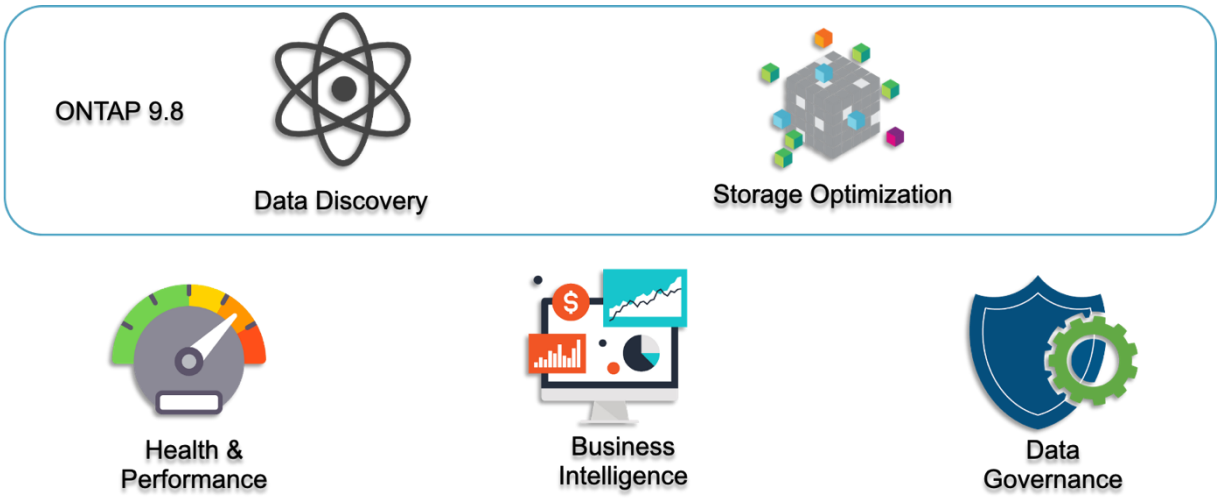
Use cases

There are five major use cases for File System Analytics.

- Data discovery
- Storage optimization
- Health and performance
- Business intelligence
- Data governance

In ONTAP 9.8, we focus on data discovery and storage optimization.

Figure 14) ONTAP File System Analytics major use cases.



Data discovery

Data discovery is about data distribution at the directory level. Customers require information about their workloads, including the space used by a project, user identities, subfolder information, directories having more files, and the average node size. You can acquire this information with third party tools, however, doing so can be a time consuming process that affects system performance and produces inconsistent information as the filesystem changes files over time. In this case, an on-box analytics tool is the right solution.

Storage optimization

Customers want to use their storage resources in the most effective manner possible and to move data to the right storage tier based on data age. They want to locate old data that has not been modified or accessed so that the admin can move or archive the data to a secondary storage tier or cloud.

Health and performance

In the heavy workload environment, customers like to perform granular troubleshooting; identify more active directories and their trees, hot files, and hot spots; and identify top users and clients based on ownership. These details help the customer to identify and resolve performance issues and load balance workload storage controllers for better utilization of resources.

Business intelligence

The customer needs to make the right decision based on time series, historical and trending data that help the customer to generate reports for data growth in percentage, forecast future data growth, and make comparisons between data points, averages, and chargeback reports. NetApp does not provide support for business intelligence with ONTAP File System Analytics in this release, but with the help of external third-party application integration with ONTAP API, we can provide this use case support to customers.

Data governance

Data governance provides for the formal management of data assets within an organization with the help of data usage, access, and behavior patterns. Data governance also visualizes these patterns and performs user behavior anomaly detection to protect organization from threats like ransomware. NetApp does not support for data governance with ONTAP File System Analytics in this release, but, with the help of external third-party application integration with ONTAP APIs, NetApp can provide support for this use case.

ONTAP File System Analytics - Capabilities

In this section, we discuss how ONTAP File System Analytics can address customer challenges.

Least and most-changed workloads

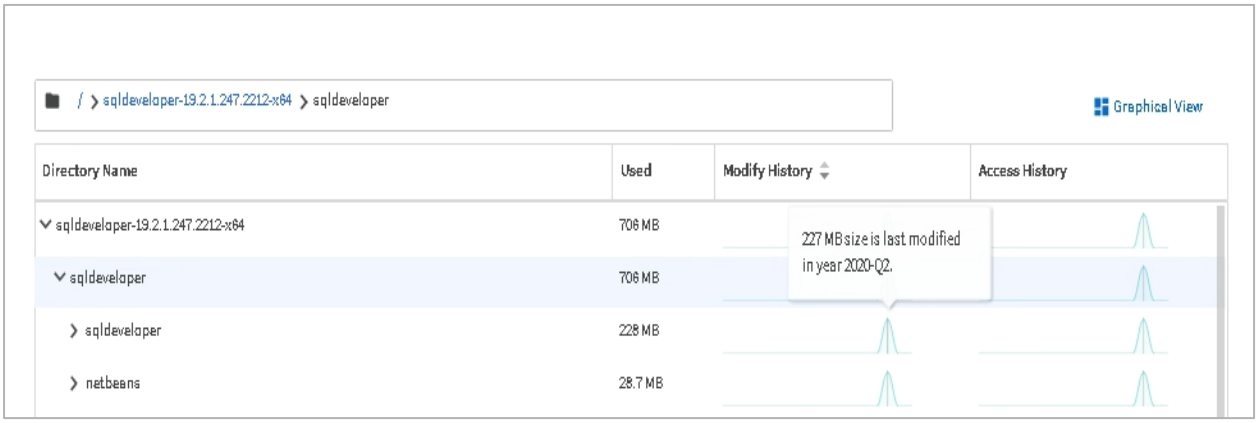
Most of the time, transactional workloads contain more changes than other workload types. To manage such workloads, customers want to see the most- and least-changed workloads and their folders.

NetApp addresses this need in two ways with ONTAP File System Analytics: the GUI and REST APIs.

GUI-based solution

The following list view describes each folder’s behavior based on modification and access history. You can sort the folders based on name, used size, modify and access history (Volume > Explorer > Modify History). The Modify History option provides the most and least changed folders in ascending or descending order. Based on the folder name, you can check the workload running on the folder.

Figure 15) System Manager view for most and least changed workloads.



REST API-based solution

Some customers like to access data using REST APIs. See the following example, which provides the least changed folders. For most changed folders, change the `order_by` to `analytics.by_modified_time.bytes_used+desc`.

```
[root@scspr1936701025 FSAFG]# curl -siku ${user}:${pw} --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volume_uuid}/files/%2F?fields=analy
tics.by_modified_time.bytes_used&order_by=analytics.by_modified_time.bytes_used+asc"
HTTP/1.1 200 OK
Date: Wed, 21 Oct 2020 20:15:04 GMT
Server: libzapid-httpd
X-Content-Type-Options: nosniff
Cache-Control: no-cache,no-store,must-revalidate
Content-Type: application/hal+json
Transfer-Encoding: chunked

{
  "records": [
    {
      "name": ".."
    },
    {
      "name": "src_million_folder_Sep091600095274.tar.gz"
    },
    {
      "name": ".snapshot"
    },
    {
      "name": "20170504_090917",
      "analytics": {
        "by_modified_time": {
          "bytes_used": {
            "values": [
              12288,
              0,
              0,
              0,
              0,
              12288,
              0,
              0,
              12288,
              0,
              0,
              0,
              12288,
              0,
              0,
```

```
            110592,
            0,
            0
          ],
          "percentages": [
            10.00,
            0.00,
            0.00,
            0.00,
            0.00,
            10.00,
            0.00,
            0.00,
            10.00,
            0.00,
            0.00,
            0.00,
            10.00,
            0.00,
            0.00,
            90.00,
            0.00,
            0.00
          ],
          "newest_label": "2020-W43",
          "oldest_label": 2017
        }
      }
    }
  },
  {
    "name": "20170504_090829",
    "analytics": {
      "by_modified_time": {
        "bytes_used": {
          "values": [
            12288,
            0,
            0,
            0,
            0,
            12288,
            0,
            0,
            12288,
            0,
            0,
            0,
            12288,
            0,
            0,
            122880,
            0,
            0
          ],
          "percentages": [
            9.09,
            0.00,
            0.00,
            0.00,
            0.00,
            9.09,
            0.00,
            0.00,
            9.09,
            0.00,
            0.00,
            0.00,
            9.09,
            0.00,
            0.00,
```



```
              90.91,
              0.00,
              0.00
            ],
            "newest_label": "2020-W43",
            "oldest_label": 2017
          }
        }
      }
    },
   …
…
<removed some results to save page spaces>
},
    {
      "name": ".",
      "analytics": {
        "by_modified_time": {
          "bytes_used": {
            "values": [
              1228800,
              0,
              0,
              0,
              224616448,
              1228800,
              5210320896,
              0,
              1228800,
              5210320896,
              739082240,
              0,
              5950631936,
              1130496,
              0,
              393216,
              0,
              0
            ],
            "percentages": [
              0.02,
              0.00,
              0.00,
              0.00,
              3.77,
              0.02,
              87.54,
              0.00,
              0.02,
              87.54,
              12.42,
              0.00,
              99.97,
              0.02,
              0.00,
              0.01,
              0.00,
              0.00
            ],
            "newest_label": "2020-W43",
            "oldest_label": 2017
          }
        }
      }
    }
  ],
  "num_records": 12,
  "analytics": {
    "by_modified_time": {
      "bytes_used": {
        "labels": [
```



```
          "2020-W43",
          "2020-W42",
          "2020-W41",
          "2020-W40",
          "2020-W39",
          "2020-10",
          "2020-09",
          "2020-08",
          "2020-Q4",
          "2020-Q3",
          "2020-Q2",
          "2020-Q1",
          "2020",
          "2019",
          "2018",
          "2017",
          "--2016",
          "unknown"
        ]
      }
    }
  },
  "_links": {
    "self": {
      "href": "/api/storage/volumes/2538a931-13c1-11eb-bfde-
005056a772a7/files/%2F?fields=analytics.by_modified_time.bytes_used&order_by=analytics.by_modifie
d_time.bytes_used+Asc"
    }
  }
}[root@scspr1936701025 FSAFG]# curl -siku ${user}:${pw} --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volume_uuid}/files/%2F?fields=analy
tics.by_modified_time.bytes_used&order_by=analytics.by_modified_time.bytes_used+Asc"  | less
[root@scspr1936701025 FSAFG]#
```

In the previous example, you can also use `subfileds` like `analytics.by_modified_time.bytes_used.values`, `analytics.by_modified_time.bytes_used.labels`, and `analytics.by_modified_time.bytes_used.percentage` in the field values.

Largest folders

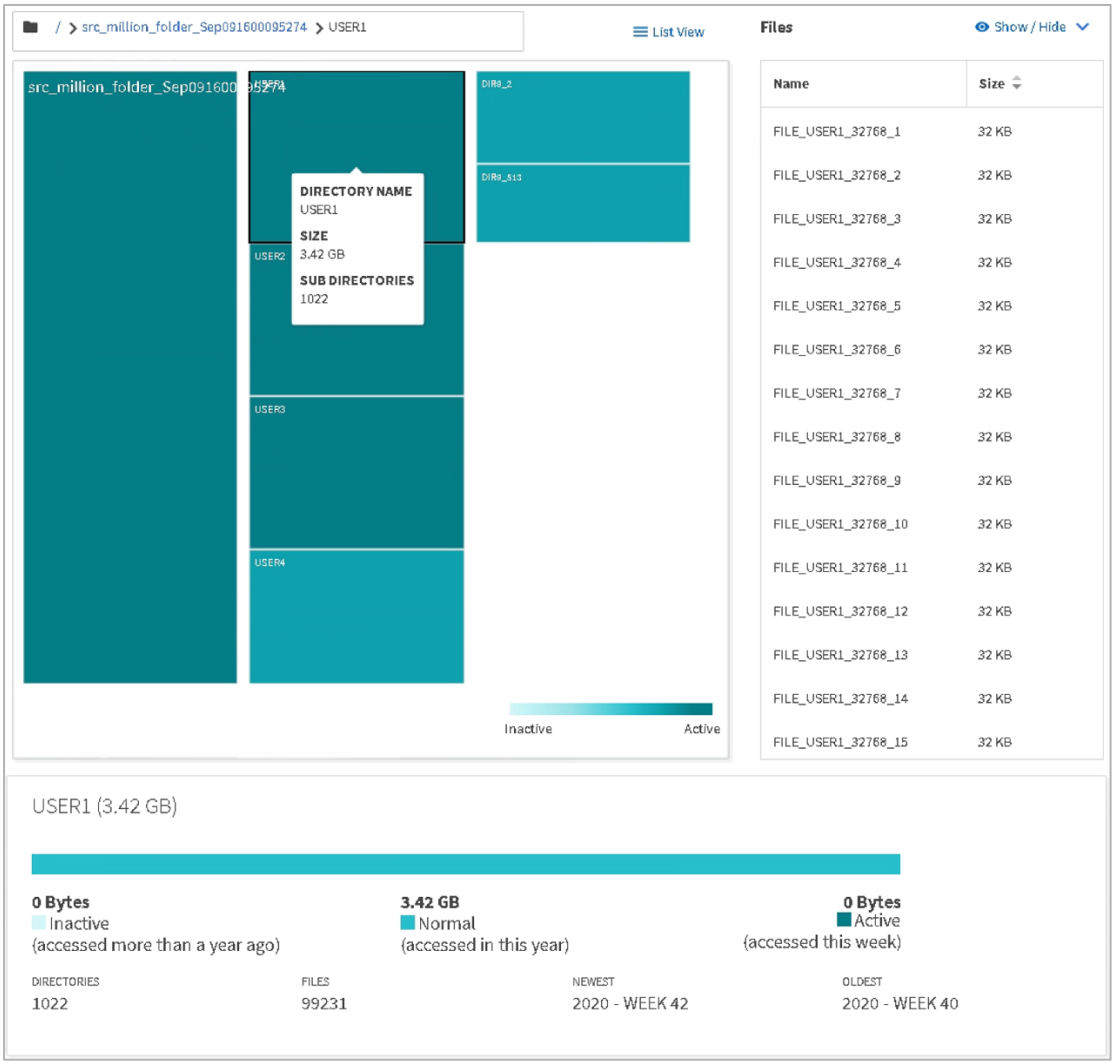
You often want to know which folder consumes the most space in you NAS volume. For example, you might want to know which users have the largest home folders.

We can provide the solution through ONTAP File System Analytics using the GUI or REST APIs.

GUI solution

NetApp ONTAP File System Analytics provides real-time analytics without external tools. You can drill down to specific folders based on their size (Volume > Explorer > Graphical View). You can also see the number of subfolders and the size of each folder through Tool Tip. For example, you can select a specific user's home folder to get more information, such as number of number of files, folders, newest and oldest files, active, normal, and inactive size. The files view also provides a list of files from a user's home folder.

Figure 16) System Manager view for the largest folders.



REST API solution

The following example depicts an option to collect the largest folders using a REST API.

```
[root@scspr1936701025 FSAFG]# curl -siku ${user}:${pw} --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volume_uuid}/files/%2F?fields=analy
tics.bytes_used&order_by=analytics.bytes_used+desc" | egrep 'name|bytes_used'
      "name": ".",
        "bytes_used": 5952155648
      "name": "src_million_folder_Sep091600095274",
        "bytes_used": 4985806848
      "name": "sqldeveloper-19.2.1.247.2212-x64",
        "bytes_used": 740012032
      "name": "20190626_162237",
        "bytes_used": 765952
      "name": "20190626_162128",
        "bytes_used": 262144
      "name": "20190626_161806",
        "bytes_used": 253952
      "name": "20170504_083851",
        "bytes_used": 172032
      "name": "20170504_090829",
        "bytes_used": 135168
```

```
      "name": "20170504_090917",
        "bytes_used": 122880
      "name": ".."
      "name": "src_million_folder_Sep091600095274.tar.gz"
      "name": ".snapshot"
      "href": "/api/storage/volumes/2538a931-13c1-11eb-bfde-
005056a772a7/files/%2F?fields=analytics.bytes_used&order_by=analytics.bytes_used+desc"
[root@scspr1936701025 FSAFG]#
```

Note: In the REST API example, `%2F` means `/`.

Locate large numbers of files

Identifying large files and folders helps you to move them to appropriate storage. Some large files contain cold data and yet consume significant space in primary storage; these folders can be moved to secondary or cloud storage. Locating a large number of files in an NAS volume helps you to select the appropriate data location.

ONTAP System Manager does not have a UI interface, But NetApp provides REST API support to get large numbers of files. See the following example to acquire a large number of files in a folder in both descending and ascending order.

```
[root@scspr1936701025 FSAFG]# curl -siku ${user}:${pw} --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volume_uuid}/files/%2F?fields=analy
tics.file_count&order_by=analytics.file_count+asc"  | egrep 'name|file_count'
      "name": ".."
      "name": "src_million_folder_Sep091600095274.tar.gz"
      "name": ".snapshot"
      "name": "20170504_083851",
        "file_count": 8
      "name": "20170504_090829",
        "file_count": 8
      "name": "20170504_090917",
        "file_count": 8
      "name": "20190626_161806",
        "file_count": 24
      "name": "20190626_162128",
        "file_count": 24
      "name": "20190626_162237",
        "file_count": 31
      "name": "sqldeveloper-19.2.1.247.2212-x64",
        "file_count": 2390
      "name": "src_million_folder_Sep091600095274",
        "file_count": 133679
      "name": ".",
        "file_count": 136173
      "href": "/api/storage/volumes/2538a931-13c1-11eb-bfde-
005056a772a7/files/%2F?fields=analytics.file_count&order_by=analytics.file_count+asc"
[root@scspr1936701025 FSAFG]# curl -siku ${user}:${pw} --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volume_uuid}/files/%2F?fields=analy
tics.file_count&order_by=analytics.file_count+desc"  | egrep 'name|file_count'
      "name": ".",
        "file_count": 136173
      "name": "src_million_folder_Sep091600095274",
        "file_count": 133679
      "name": "sqldeveloper-19.2.1.247.2212-x64",
        "file_count": 2390
      "name": "20190626_162237",
        "file_count": 31
      "name": "20190626_161806",
        "file_count": 24
      "name": "20190626_162128",
        "file_count": 24
      "name": "20170504_083851",
        "file_count": 8
      "name": "20170504_090829",
        "file_count": 8
      "name": "20170504_090917",
```

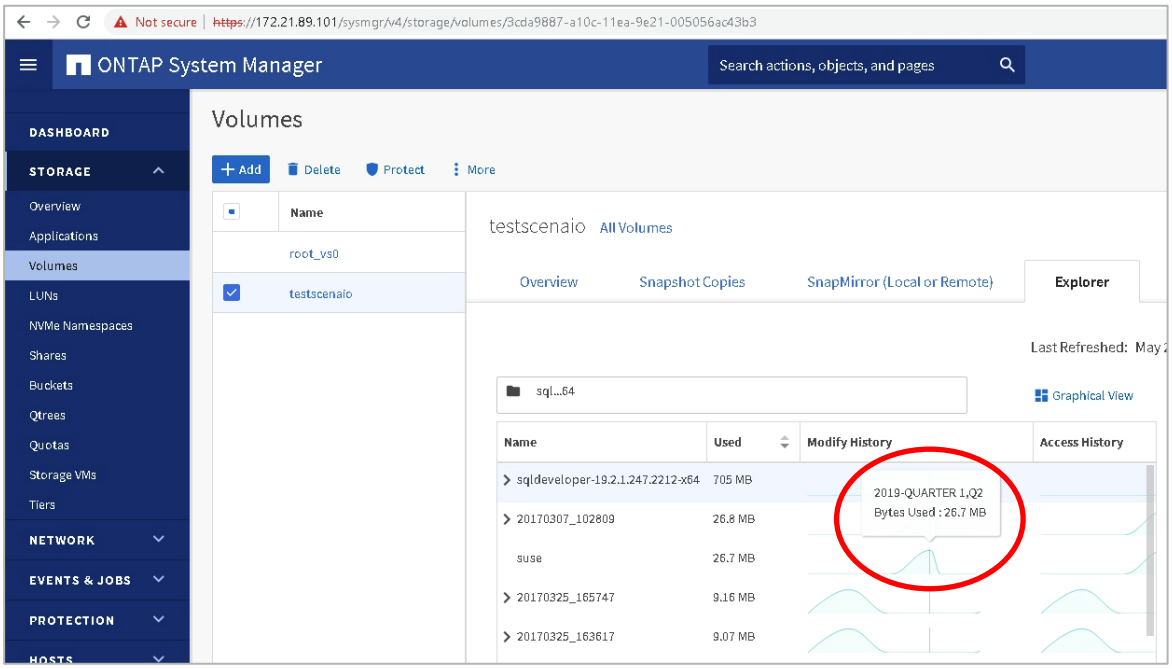
```
        "file_count": 8
      "name": ".."
      "name": "src_million_folder_Sep091600095274.tar.gz"
      "name": ".snapshot"
      "href": "/api/storage/volumes/2538a931-13c1-11eb-bfde-
005056a772a7/files/%2F?fields=analytics.file_count&order_by=analytics.file_count+desc"
[root@scspr1936701025 FSAFG]#
```

Activity behavior

To identify aberrant behavior in sensitive data, you can detect unexpected spikes in access or other parameters. For example, a ransomware attack might affect some folders (Volume > Folder > Folder List > select the folder that has abnormal spikes), and the admin can move these folders and their data to alternative secondary storage for further investigation.

You can monitor activity using the GUI as follows:

Figure 17) System Manager aberrant activity detection.



JSON format of analytics REST API

Table 3 lists the analytics REST API JSON format and their descriptions. It helps the developer to design their application based on Analytics API.

Table 3) Analytics REST API JSON.

Field	Subfield	Description
Name		Name of the directory entry
File_count		Files count in the root of the directory
Bytes_used		Size of the directory
by_modified_time		Modification history
	bytes used	Changed data size in last (beyond 4 years, year 3, years 2, Month4, Month3)

Field	Subfield	Description
	percentages	Percentage of data modified out of total directory size
	newest folder	Latest folder name which is modified
	oldest folder	Oldest directory
Access history		Accessed history
	bytes used	Size of data accessed in last (beyond 4 years, year 3, year 2, Month4, months3)
	percentages	Percentage of data modified out of total directory size
	newest folder	Latest folder name which was accessed
	oldest folder	Oldest directory – not touched for more than a year

Conclusion

ONTAP File System Analytics provides the following benefits:

- Mapping and characterizing file structure, including identifying the most or least changed data, identifying the largest folders, and determining where the largest number of files are located
- Identifying abnormal data behavior
- Real-time analytics
- Application integration with ONTAP File System Analytics through REST APIs
- Support for the data-discovery and storage-optimization use cases
- The summary panel identifies hot (active), cold (inactive), and warm (normal) data
- Data sorting based on modification and access time
- Cost-effective decision making for primary and secondary storage
- Granular directory access

Appendix

```
[root@scspr1936701025 ~]# cat apitemplate.sh
#! /usr/software/bin/bash

path_to_chars()
{
        result="$1"
        count=`echo ${result}|wc -m`
        result_temp=''
        for cursor in `seq 1 $count`
        do
                c1=`echo $result|cut -c${cursor}`
                if test "$c1" = "/"
                then
                        c1="%2F"
                fi
                result_temp+=${c1}
        done
        result=${result_temp}
        eval $2='$result'
}

echo -n "User : "
read user
[[ -z $user ]] && { echo "User name is empty"; exit 1; }

echo -n "Password : "
```

```
read -s pw
echo
[[ -z $pw ]] && { echo "Password is empty"; exit 1; }

echo -n "ONTAP Cluster Management IP/Name : "
read server
[[ -z $ontap_cluster_management_ip ]] && { echo "ONTAP Cluster Management ip or name is empty";
exit 1; }

echo -n "Volume UUID : "
read volumeuuid;
[[ -z $volumeuuid ]] && { echo "volume uuid is empty"; exit 1; }

echo "choose the API from the list if not enter the API name:"

echo -e " 1. analytics.file_count"
echo -e " 2. analytics.by_modified_time.bytes_used.oldest_label"
echo -e " 3. analytics.by_modified_time.bytes_used.percentages"
echo -e " 4. analytics.by_modified_time.bytes_used.values"
echo -e " 5. analytics.by_modified_time.bytes_used.oldest_label"
echo -e " 6. analytics.by_modified_time.bytes_used.newest_label"
echo -e " 7. analytics.by_modified_time.bytes_used.labels"
echo -e " 8. analytics.bytes_used"
echo -e " 9. analytics.subdir_count"
echo -e " 10. analytics.by_accessed_time.bytes_used.oldest_label"
echo -e " 11. analytics.by_accessed_time.bytes_used.percentages"
echo -e " 12. analytics.by_accessed_time.bytes_used.values"
echo -e " 13. analytics.by_accessed_time.bytes_used.newest_label"
echo -e " 14. analytics.by_accessed_time.bytes_used.labels"
echo -e " 0. Others"

echo -n "choose the number or '0' for provide API name : "
read apiname
[[ -z $apiname ]] && { echo "apiname is empty"; exit 1; }

#echo $apiname
case $apiname in
        "1")
                apiname="analytics.file_count"; echo "API choosen : $apiname"
                ;;
        "2")
                apiname="analytics.by_modified_time.bytes_used.oldest_label"; echo "API choosen :
$apiname"
                ;;
        "3")
                apiname="analytics.by_modified_time.bytes_used.percentages"; echo "API choosen :
$apiname"
                ;;
        "4")
                apiname="analytics.by_modified_time.bytes_used.values"; echo "API choosen :
$apiname"
                ;;
        "5")
                apiname="analytics.by_modified_time.bytes_used.oldest_label"; echo "API choosen :
$apiname"
                ;;
        "6")
                apiname="analytics.by_modified_time.bytes_used.newest_label"; echo "API choosen :
$apiname"
                ;;
        "7")
                apiname="analytics.by_modified_time.bytes_used.labels"; echo "API choosen :
$apiname"
                ;;
        "8")
                apiname="analytics.bytes_used"; echo "API choosen : $apiname"
                ;;
        "9")
                apiname="analytics.subdir_count"; echo "API choosen : $apiname"
                ;;
        "10")
```



```
                apiname="analytics.by_accessed_time.bytes_used.oldest_label"; echo "API choosen :
$apiname"
                ;;
        "11")
                apiname="analytics.by_accessed_time.bytes_used.percentages"; echo "API choosen :
$apiname"
                ;;
        "12")
                apiname="analytics.by_accessed_time.bytes_used.values"; echo "API choosen :
$apiname"
                ;;
        "13")
                apiname="analytics.by_accessed_time.bytes_used.newest_label"; echo "API choosen :
$apiname"
                ;;
        "14")
                apiname="analytics.by_accessed_time.bytes_used.labels"; echo "API choosen :
$apiname"
                ;;
        "0")
                echo -n " Enter the API Name : "
                read apiname
                echo "API name : $apiname"
                ;;
        *)
                echo "Entries not exist";
                exit 1;
                ;;
esac

echo -n "Full path. default(/): "
read path_input;
[[ -z $path_input ]] && { path_input="/"; }
path_to_chars $path_input path

echo -n "Do you want to call the API with default options. (y/n): default(y):"
read default_yes_or_no;
[[ -z $default_yes_or_no ]] && { default_yes_or_no="y"; }
default_yes_or_no="${default_yes_or_no,,}"

if test $default_yes_or_no = "y" -o $default_yes_or_no = "yes"
then
        #API call to ontap filesystem analytics
        echo "curl -siku $user:xxx  --request GET
\"https://${ontap_cluster_management_ip}/api/storage/volumes/${volumeuuid}/files/${path}?fields=$
{apiname}\" "
        curl -siku $user:$pw  --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volumeuuid}/files/${path}?fields=${
apiname}"
else
        #order_by : field  - filedname. default is API name
        echo -n "order_by filedname. default is API name($apiname): "
        read order_by_fieldname;
        [[ -z $order_by_fieldname ]] && { order_by_fieldname=$apiname; }

        #order_by : order_by. default assending order
        echo -n "order_by. asc/desc : default (asc): "
        read order_by;
        order_by="${order_by,,}"
        [[ -z $order_by ]] && { order_by="asc"; }

        #initialize extra options
        extra_options='';
        #return_timeout - default 15 sec
        echo -n "Number of seconds to allow the call to execute before returning: default (15): "
        read return_timeout;
        [[ -z $return_timeout ]] && { return_timeout=15; }
        if test $return_timeout -ne 15
        then
                extra_options+="&return_timeout=${return_timeout}"
        fi
```



```
        #API call to ontap filesystem analytics
        echo "user pw ontap_cluster_management_ip volumeuuid apiname path fieldname order_by
return_records max_records return_timeout"
        echo $user xxxx $ontap_cluster_management_ip $volumeuuid $apiname $path
$order_by_fieldname $order_by $return_records $max_records $return_timeout
        echo "curl -siku ${user}:xxx  --request GET
\"https://${ontap_cluster_management_ip}/api/storage/volumes/${volumeuuid}/files/${path}?fields=$
{apiname}&order_by=${order_by_fieldname}+${order_by}${extra_options}\" "
        curl -siku $user:$pw  --request GET
"https://${ontap_cluster_management_ip}/api/storage/volumes/${volumeuuid}/files/${path}?fields=${
apiname}&order_by=${order_by_fieldname}+${order_by}${extra_options}"
fi
[root@scspr1936701025 ~]#
```

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp Product Documentation
  https://docs.netapp.com

Version history

Version	Date	Document Version History
Version 1.0	October 2020	Initial release
Version 1.1	April 2023	• ONTAP 9.13.1 • FSA initialization scan section

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

TR-4867-1121

NetApp