POC - Media Shuttle

Overview

AmES smartflow automation requires a common datastore (S3 or Shared File System). Interaction with this data store is paramount for its success. We need auto file copy when user travels between processes, file/folder level permission control, upload/download while working on a specific task in the workflow. S3 as a datastore was ruled out due to high cost of operation, although its a lot faster than local file copy over network. Mediashuttle (MS) along with network file copy is being used by production users. It supports LocalFileSystem and cloud services and it accelarates file copy over the netowrk. In this exercise, we will test if MS is faster than local file copy over the network. We will also test out MS's Restful APIs.

Areas tested

1. upload files and folders manually
2. download files and folders manually
3. upload files and folders through MS restful API integration
4. download files and folders through MS restful API integration
5. versioning of files on each upload
6. track upload/download progress through MS API
7. copy/move files or folders as a background activity through MS API
8. transfer file within system through MS REST API
9. register a user/email through MS REST API
10. check history of a package id through MS REST API
11. check current active transfers through MS REST API
12. create a new portal through MS REST API
13. set user roles or persmission through MS
14. storage replication/backup feature

Result

S.No	Task	Status
1.	upload files and folders manually	**Yes**
2.	download files and folders manually	**Yes**
3.	upload files and folders through MS restful API integration	No
4.	download files and folders through MS restful API integration	No
5.	versioning of files on each upload	No
6.	track upload/download progress through MS API	**Yes**
7.	copy/move files or folders as a background activity through MS API	No
8.	transfer file within system through MS REST API	No

S.No	Task	Status
9.	register a user/email through MS REST API	**Yes**
10.	check history of a package id through MS REST API	**Yes**
11.	check current active transfers through MS REST API	**Yes**
12.	create a new portal through MS REST API	**Yes**
13.	set user roles or persmission through MS	**Yes**
14.	storage replication/backup feature	**No**

Trend Download & Upload

S.No	Type	Size	Mode	Duration (Minutes)
1.	Upload	250 Mb	Without - Agent	0:02:17
2.	Upload	250 Mb	With - Agent	0:01:44
3.	Upload	3.5 Gb	Without - Agent	0:22:01
4.	Upload	3.5 Gb	With - Agent	0:20:01
5.	Download	250 Mb	Without - Agent	0:03:17
6.	Download	250 Mb	With - Agent	0:02:17
7.	Download	3.5 Gb	Without - Agent	0:14:50
8.	Download	3.5 Gb	With - Agent	0:14:01

Pros

- user-friendly upload and download interface
- process webhook for process status
- the unlimited size of upload
- on-premises and cloud datastore

Cons

- The download with and without the app(agent) are comparatively equal
- The upload with and without the app(agent) are comparatively equal
- Cannot upload/download a file on the Linux platform
- In case of failure, the fallback process has to think and plan
- there is no direct API for copying a file from one folder to another. If we want, have to go for a jet workflow system.
- Replication/backup information not available

Conclusion

As per the report, MS has scored 57% for the usecases, and rest of 43% have to plan go for alternative approach. Though it has some upsides it is also have more downsides. Some of Major feature does not available on REST API. Trend of Download and upload with with-agent and without-agent are not impressive both are looking same. So, I recommend we will go for local FileSysten service.

Exercises

Point 1 - manual upload

> Yes, can upload files and folders manually by creating an upload request (link)

> Steps

1. create an empty package id
    1. required fields are `portal_id`
2. create an upload URL
    1. required fields are
        1. `portal_id`
        2. `package_id`
        3. grants `upload`
        4. `email`
    2. optional fields are
        1. `destinationPath`
        2. `webhook`
3. conditions
    1. `email` should be registered/activated on the media shuttle portal
    2. cannot create multiple `upload` requests on a single package id
    3. the user has to login into the portal to upload files
    4. Once the link has been used it expires automatically
    5. Once files are uploaded the files will be placed under `destinationPath` if given in the payload otherwise it is placed as `root path`
    6. If the same files are present inside the directory then they will be overwritten

---

Point 2 - manual download

> Yes, we can download files and folders manually by creating a download files request(link)

> Steps

1. create a download URL
    1. required fields are
        1. `portal_id`
        2. `package_id` - existing
        3. grants `download`
        4. `email`
    2. optional fields are

        1. `webhook`
    2. conditions
        1. `email` should be registered/activated on the media shuttle portal
        2. can create multiple `download` requests on a single package id
        3. the user has to login into the portal to upload files
        4. Once the link has been used it expires automatically
        5. the package id's associated files are only displayed on the download window

Point 3 upload files by automation

> No, there is no option to upload a file directly through the rest API.
>
> But, still there are possibilities.

> Steps

1. move or copy the files directly through the SCDX server path.
2. have to execute a script inside the fileServer

Import:

2. has to associate files into a `package-id`. Then can share these files at in download link
3. If the same files are present inside the directory then they will be overwritten

Point 4 download files by automation

> No, there is no option to upload the file directly through the rest API.
>
> But, still there are possibilities.

> Steps

1. move or copy the files directly through the SDCX server path.
2. have to execute/run a script inside the fileServer

Point 5 file versioning

> No, there is no option to file versioning each upload on the same file it will overwrite the existing one

Point 6 Process state

> Yes, can receive process state on success as well as failure cases through `webhook`

> Steps

1. have to set a `webhook` on `upload` or download request link

Point 7 copy folder to folder

> No, there is no rest API found for folder copy but, can do it manually.

Import:

1. have to associate files into a `package-id`.Then we can share these files in the download link