Eric Feldman
CSS 436 – Cloud Computing
Assignment 3 – Cloud Backup
October 24, 2019

Application, Design, and Test

Application and Design

For program 3, a python application was made to create a backup of a users' data to a cloud service (in this case using ASW S3). The python backup program was primary broken up into 5 different methods. These are: `upload(fileName, s3folder, name)`, `walkDirectory()`, `walkMemoryCheck()`, `readMetaData()`, `writeMetaData()`. The *upload* method accepts a file name, s3 folder, and a name. This method then proceeds to handle new and existing directories. It parses the path names and makes calls to AWS S3 using boto3. In addition, files are checked if there have been modifications to them, and only pushes files to AWS S3 if changes have been made to reduce data bandwidth. A global dictionary called *mem* is used to keep track of all the files that are in the local directory and have passed through upload. The *walkDirectory()* method is used to 'walk' through the local directory and make calls to the upload helper method to handle every file that is encountered. The method *walkMemoryCheck* is used to go through the mem dictionary to check if any files have been deleted from the local directory, if they have then it will be removed from the metadata and from AWS S3 bucket. This way the program is able to ensure that the AWS S3 backup and local directory always stay in sync. The method *readMetaData* is used to read in the metadata text file in the local directory. If a metadata file already exists then it is read into the mem dictionary, otherwise a new metadata file is opened. Lastly, the method *writeMetaData* goes through the mem dictionary and writes the metadata to the text document.

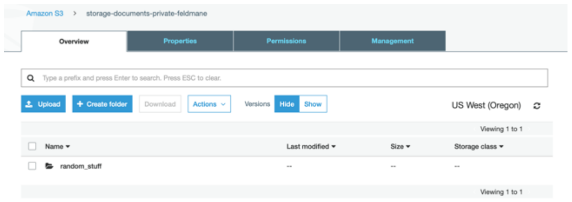
In order to ensure security the AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, and BUCKET_NAME are all read into the program from an external text file in the same directory as the backup.py file.

Testing

Various tests were completed to ensure all files would backup correctly, in the same file structure, and only changed files would backup if they already existed previously (to reduce data bandwidth).

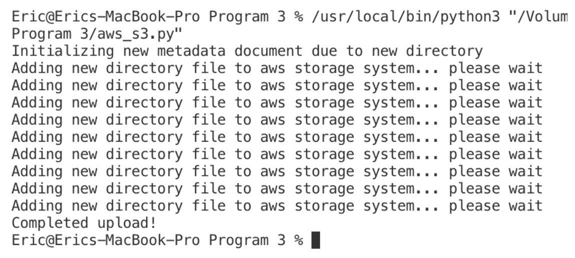
<u>Test 1</u>

The first test involved creating a new backup of a directory that has never been backed up using this program.



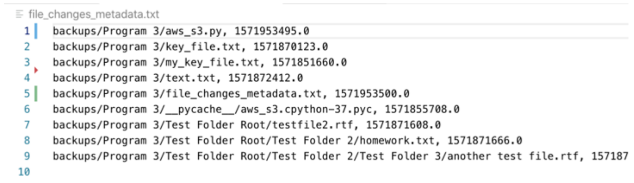
Name	Last modified	Size	Storage class
random_stuff	--	--	--

We can see in this image that currently my one of my root S3 buckets has only one sub-bucket called random_stuff with no backup in it.



```
Eric@Erics-MacBook-Pro Program 3 % /usr/local/bin/python3 "/Volum
Program 3/aws_s3.py"
Initializing new metadata document due to new directory
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Adding new directory file to aws storage system... please wait
Completed upload!
Eric@Erics-MacBook-Pro Program 3 %
```

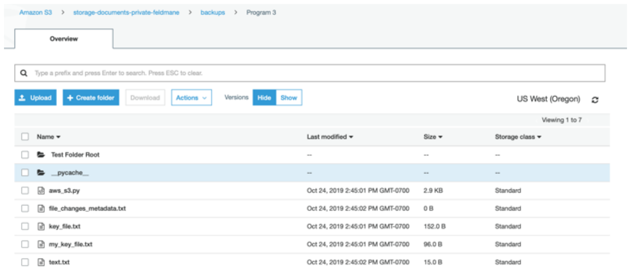
After running the program for the first time we can see the console output here. It is clear that the program interpreted this as a new directory and preceded to create a new metadata document and proceeded to upload all the files.



file_changes_metadata.txt

```
backups/Program 3/aws_s3.py, 1571953495.0
backups/Program 3/key_file.txt, 1571870123.0
backups/Program 3/my_key_file.txt, 1571851660.0
backups/Program 3/text.txt, 1571872412.0
backups/Program 3/file_changes_metadata.txt, 1571953500.0
backups/Program 3/__pycache__/aws_s3.cpython-37.pyc, 1571855708.0
backups/Program 3/Test Folder Root/testfile2.rtf, 1571871608.0
backups/Program 3/Test Folder Root/Test Folder 2/homework.txt, 1571871666.0
backups/Program 3/Test Folder Root/Test Folder 2/Test Folder 3/another test file.rtf, 157187
```

Here we can see that the correct file structure was maintained for the directory being uploaded. A backups folder is created within the bucket name that was provided by the user in the text file. Every file path is followed by a comma and the time in seconds when the file was last modified.

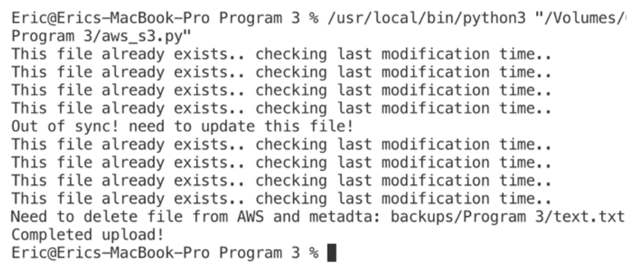

Amazon S3 > storage-documents-private-feldmane > backups > Program 3
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
Versions
Hide
Show
US West (Oregon)
Viewing 1 to 7

Name	Last modified	Size	Storage class
Test Folder Root	--	--	--
__pycache__	--	--	--
aws_s3.py	Oct 24, 2019 2:45:01 PM GMT-0700	2.9 KB	Standard
file_changes_metadata.txt	Oct 24, 2019 2:45:02 PM GMT-0700	0 B	Standard
key_file.txt	Oct 24, 2019 2:45:01 PM GMT-0700	152.0 B	Standard
my_key_file.txt	Oct 24, 2019 2:45:01 PM GMT-0700	96.0 B	Standard
text.txt	Oct 24, 2019 2:45:02 PM GMT-0700	15.0 B	Standard

A folder called backups was created and shows all the files that been successfully uploaded to S3 with the correctly maintained file structure.

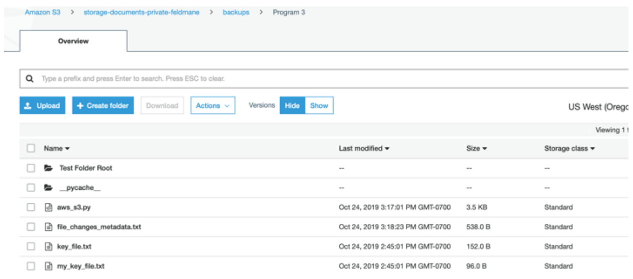
<u>Test 2</u>

The second test involved deleting a file in a directory that has been previously uploaded using the backup program.



```
Eric@Erics-MacBook-Pro Program 3 % /usr/local/bin/python3 "/Volumes/
Program 3/aws_s3.py"
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
Out of sync! need to update this file!
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
Need to delete file from AWS and metadta: backups/Program 3/text.txt
Completed upload!
Eric@Erics-MacBook-Pro Program 3 %
```

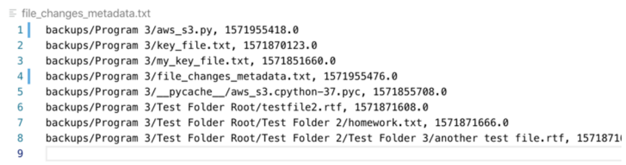
Here we can see the program correctly realized that a file was deleted from the local directory and needed to update the metadata and AWS S3 directory



Amazon S3 > storage-documents-private-feldmane > backups > Program 3
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
+ Create folder
Download
Actions
Versions
Hide
Show
US West (Orego
Viewing 1

Name	Last modified	Size	Storage class
Test Folder Root	--	--	--
__pycache__	--	--	--
aws_s3.py	Oct 24, 2019 3:17:01 PM GMT-0700	3.5 KB	Standard
file_changes_metadata.txt	Oct 24, 2019 3:18:23 PM GMT-0700	538.0 B	Standard
key_file.txt	Oct 24, 2019 2:45:01 PM GMT-0700	152.0 B	Standard
my_key_file.txt	Oct 24, 2019 2:45:01 PM GMT-0700	96.0 B	Standard

This image clearly shows that the file text.txt was deleted from the local directory because it no longer exists in the AWS S3 directory either.

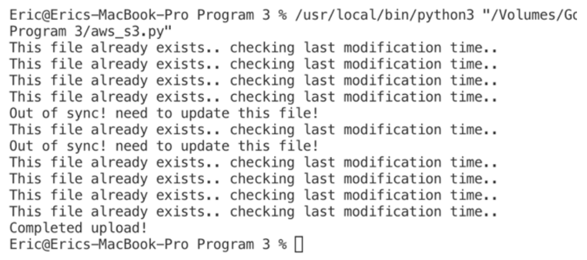


```
1  backups/Program 3/aws_s3.py, 1571955418.0
2  backups/Program 3/key_file.txt, 1571870123.0
3  backups/Program 3/my_key_file.txt, 1571851660.0
4  backups/Program 3/file_changes_metadata.txt, 1571955476.0
5  backups/Program 3/__pycache__/aws_s3.cpython-37.pyc, 1571855708.0
6  backups/Program 3/Test Folder Root/testfile2.rtf, 1571871608.0
7  backups/Program 3/Test Folder Root/Test Folder 2/homework.txt, 1571871666.0
8  backups/Program 3/Test Folder Root/Test Folder 2/Test Folder 3/another test file.rtf, 1571871
9
```

We can see the metadata got updated and removed the file text.txt. There is also now 8 lines instead of the previous 9 lines.

<u>Test 3</u>

The second test involved modifying a file in a directory that has been previously uploaded using the backup program.



```
Eric@Erics-MacBook-Pro Program 3 % /usr/local/bin/python3 "/Volumes/Gc
Program 3/aws_s3.py"
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
Out of sync! need to update this file!
This file already exists.. checking last modification time..
Out of sync! need to update this file!
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
This file already exists.. checking last modification time..
Completed upload!
Eric@Erics-MacBook-Pro Program 3 %
```

Here we can see the program correctly interpreted a file that was modified and only updated that file in the AWS S3 storage. If one file was modified then there will always be at least two new files being uploaded because the second file that will become out of sync is the metadata file.