Mitchell Kane

2/14/2020

CSS 436 Cloud Computing

Dimpsey

Program 3

This program is intended to copy local files to an AWS s3 bucket, as well as restoring those files by copying them from the bucket back to your local machine. During the backup process, files will only be copied if they have been modified at some point since they were last backed up to the bucket. Also, during this upload process, the structure and relationship of files on your local machine is maintained within the bucket.

While downloading files to and from the bucket, files will be copied with the same namespace and relationship to each other as exists in the bucket. This namespace and relationship is maintained in the local folder destination. This program utilizes a premade bucket with public access as demonstrated during CSS436 class, so my personal private and secret keys are not provided or needed. If access was restricted, I would have provided those secret keys in a text file and included a method in my program to create access for whoever was running this program.

These programs were created and tested in VS Code on my Macintosh, using python 2.7, pip3 20.0.2, and boto3. In this zip folder I provided a folder (tree) of files to backup to the s3 bucket, “FilesToUpload” and have made the s3 bucket empty initially. Wherever you place this program, it will upload any directory or file that is in the same directory, or a subdirectory of it. After executing the backup, you will see the files in the bucket with the same structure and relationship. The program does check to ensure the directory name the user enters exists somewhere in the directory of the program, and it can handle relative or absolute path names.

The Amazon site can be tricky. You will see a file named “.” Which represents the working directory of the program. Click on the period once, nothing will happen, this is ok. Above the period there are multiple tabs, click “Properties,” then click “Overview.” Now you will see a file named “FilesToUpload.” If you click on that folder you will observe the same files I have provided, and you have locally.

For the restore part of this program, I encourage you to remove the text files and folders from your local disk, this will offer proof the download was successful. If the folders exist, the program will restore the files to it. Otherwise a directory will be created to store the file in.

To run the backup program, in your terminal application ‘cd’ to the program location and type:

python Backup.py ./FilesToUpload

or

python Backup.py FilesToUpload

or

python Backup.py <your subdirectory>

or

python Backup.py <absolute file path>

The first time this is run, the console will output each file that is uploaded. If it is run again, it will output that a file was not upload, as no files were modified since the last upload. Were you to change some content within one of the text files, running the backup program again will output the name of that modified file. If the argument (directory to upload) is misspelled or does not exists, and error message will appear.

The run the restore program, in your terminal application ‘cd’ to the program location and type:

python Restore.py <tree or directory or subdirectory you wish>

for example:

python Restore.py filefolder2

When this is run, the program will only download files in the bucket matching that directory, and the console will output the name of each file downloaded as its filepath. Should you examine your own file system, those files should be in those locations.