

How to run

Backup

- `python prog3.py backup "directory-name" "bucketname::directory-name"`

Restore

- `python prog3.py restore "bucketname::directory-name" "directory-name"`

Use quotes if a directory has a space or you are using back slash on Mac/Linux, if not then they're optional.

Please use forward slashes instead of back slashes. My program will convert all back slashes to forward slashes. Back slashes are ignored if you are using Mac OS or Linux unless you wrap them in quotes, but Windows treats it as a forward slash.

Example:

- `python prog3.py backup test\test1 bucketname::directory-name`
 - If ran on Mac or Linux the backslash will be ignored and become testtest1, to prevent that, either use forward slash or use quotes.

Example Screenshots

Ran on Windows 10.

Backups

Initial Run

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py backup test css436tonytest::test
Starting backup....
Bucket does not exist, creating bucket: css436tonytest...
Backed up test/test1/testfile1.1.txt to s3://css436tonytest/test/test1/testfile1.1.txt successfully
Backed up test/test1/testfile1.2.txt to s3://css436tonytest/test/test1/testfile1.2.txt successfully
Backed up test/test2/test2.1/test2.1.1.txt to s3://css436tonytest/test/test2/test2.1/test2.1.1.txt successfully
Backed up test/test3/test2.1/test2.1.1.txt to s3://css436tonytest/test/test3/test2.1/test2.1.1.txt successfully
Done
```

Backing up again with no changes

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py backup test css436tonytest::test
Starting backup....
Did not need to back up test/test1/testfile1.1.txt to s3://css436tonytest/test/test1/testfile1.1.txt
Did not need to back up test/test1/testfile1.2.txt to s3://css436tonytest/test/test1/testfile1.2.txt
Did not need to back up test/test2/test2.1/test2.1.1.txt to s3://css436tonytest/test/test2/test2.1/test2.1.1.txt
Did not need to back up test/test3/test2.1/test2.1.1.txt to s3://css436tonytest/test/test3/test2.1/test2.1.1.txt
Done
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> █
```

Backing up again with one change

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py backup test css436tonytest::test
Starting backup.....
Backed up test/test1/testfile1.1.txt to s3://css436tonytest/test/test1/testfile1.1.txt successfully
Did not need to back up test/test1/testfile1.2.txt to s3://css436tonytest/test/test1/testfile1.2.txt
Did not need to back up test/test2/test2.1/test2.1.1.txt to s3://css436tonytest/test/test2/test2.1/test2.1.1.txt
Did not need to back up test/test3/test2.1/test2.1.1.txt to s3://css436tonytest/test/test3/test2.1/test2.1.1.txt
Done
```

Backing up with an invalid local directory

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py backup te css436tonytest::test
te is not a valid directory
Done
```

Backing up to a bucket that doesn't exist (creates bucket)






```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py backup test css436tonytest1::test
Starting backup.....
Bucket does not exist, creating bucket: css436tonytest1....
Backed up test/test1/testfile1.1.txt to s3://css436tonytest1/test/test1/testfile1.1.txt successfully
Backed up test/test1/testfile1.2.txt to s3://css436tonytest1/test/test1/testfile1.2.txt successfully
Backed up test/test2/test2.1/test2.1.1.txt to s3://css436tonytest1/test/test2/test2.1/test2.1.1.txt successfully
Backed up test/test3/test2.1/test2.1.1.txt to s3://css436tonytest1/test/test3/test2.1/test2.1.1.txt successfully
Done
```

| | | |
|-----------------------|-----------------|----------------------------|
| <input type="radio"/> | css436tonytest | US West (Oregon) us-west-2 |
| <input type="radio"/> | css436tonytest1 | US West (Oregon) us-west-2 |

Restore

Restoring a directory from the cloud onto a directory that doesn't exist

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py restore css436tonytest::test/ fromcloud/
Restoring s3://css436tonytest/test/test1/testfile1.1.txt to fromcloud/
Restoring s3://css436tonytest/test/test1/testfile1.2.txt to fromcloud/
Restoring s3://css436tonytest/test/test2/test2.1/test2.1.1.txt to fromcloud/
Restoring s3://css436tonytest/test/test3/test2.1/test2.1.1.txt to fromcloud/
Done
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3>
```

| Name | Date modified | Type | Size |
|---|-------------------|-------------------|--------|
|  .idea | 2/21/2023 3:16 PM | File folder | |
|  fromcloud | 2/21/2023 3:17 PM | File folder | |
|  test | 2/21/2023 3:16 PM | File folder | |
|  css436_program3 (1).pdf | 2/17/2023 5:03 PM | Chrome HTML Do... | 163 KB |
|  prog3.py | 2/20/2023 9:56 PM | JetBrains PyCharm | 6 KB |

Restoring from a directory that doesn't exist

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py restore css436tonytest::tt/ fromcloud/
tt/ does not exist in css436tonytest
Done
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> 
```

Restoring from a bucket that doesn't exist

```
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> python prog3.py restore css436tonytt::test/ fromcloud/
Bucket: css436tonytt does not exist
Done
PS C:\Users\Tony\Desktop\UW School\CSS436\prog3> 
```

Design of program

The program takes in two directories from the user, one from the user's local directory and one for the cloud.

Functions:

- backup(local_path, bucket_path)
 - This function performs a backup of the specified directory (local_path) and moved the data into an S3 bucket (bucket_path). If the bucket provided doesn't exist for the user, it'll create one and then store the information in it. It'll then check to make sure that the last modified date of the local directory is later than the one in the cloud. If it is, then it'll update the one in the cloud, if not then there is no need to back up to the cloud again.
- restore(local_path, bucket_path)
 - This function performs a restore from the specific directory in S3 and then restores the data to the user's local directory. If the bucket doesn't exist then it'll just exit. It'll get all the objects up to 1000 and restore them to the user's local directory. It'll also keep the same file structure, if a file already exists it'll overwrite. If the user tries to restore to a directory that doesn't exist, it'll create it.
- download_to_local(client, local_path, bucket, key)

- Helper function for restore. This function restores a file using info passed in. If a directory doesn't exist it'll create it to maintain same structure.
- `upload_to_bucket(client, local_path, bucket, key)`
 - Helper function for backup. This function uploads a file from local directory to the cloud.
- `bucket_exist(client, bucket_name)`
 - This function just checks if a specific bucket exists. Returns true or false.