**ELEN 6770 Project**

Team member: Rui Zhang UNI:rz2561

Zhongcheng Yi UNI:zy2483

**1.Requirements**

• AWS S3 Configured

• Python 3.7+(boto3,botocore)

**2.Introduction**

We create a S3 Backup & Restore application called ‘4D-Pocket’ that can upload files from your local directory to AWS S3 cloud or restore a directory from the S3 cloud to your local directory. (Functions: backup and restore local files)

**3. AWS credentials**

Before using this program, you will need to set up your AWS credentials:

(1) Create a virtual environment for this program.

(2) Set the AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY

environment variables.

To set these variables on Linux, macOS, or Unix, use :

export AWS\_ACCESS\_KEY\_ID=your\_access\_key\_id

export AWS\_SECRET\_ACCESS\_KEY=your\_secret\_access\_key

To set these variables on Windows, use :

set AWS\_ACCESS\_KEY\_ID=your\_access\_key\_id

set AWS\_SECRET\_ACCESS\_KEY=your\_secret\_access\_key

**4.Backup**

Backup method takes in user arguments, s3 object, and s3 session:

(1) Check if the given bucket from arguments exists in the user’s buckets. If the bucket does not exist, attempt to create the bucket, otherwise exit program.

(2) Create an OS Path with directory path from arguments and an array to keep track of parent folder names, which helps us mirror the local directory on our cloud upload.

(3) Iterate through every file in local directory. Check if the file is new, or already exists on the cloud. Upload the file if it’s a different modified date since last upload.

**5.Restore**

Restore method takes in command line arguments and S3 object. Restore will download files directory from S3, and store to local directory path.

(1) Check if a given bucket name exists in user’s bucket list. If not, exit program.

(2) Iterate through the buckets starting from the directory given in the user arguments.

(3) For every element, if it’s a folder, do not attempt to download, but instead continue until we find a file to download.

**6.Run 4D-Pocket.py**

You can start the program by running 4D-Pocket.py through your IDE or using command line.

1. To Backup, run 4D-Pocket.py, and type in:

backup local-directory-name s3-bucket-name::s3-directory-name

• The local directory needs to be the full system path.

• The bucket name needs to be your S3 bucket name or new bucket name.

• Directory name will be a folder created in your S3 bucket or updated over an existing one.

(2) To restore, run 4D-Pocket.py, and type in:

restore s3-bucket-name::s3-directory-name local-directory-name

• The bucket name needs to be your S3 bucket name and directory name should exist.

• The local directory needs to be the full system path.

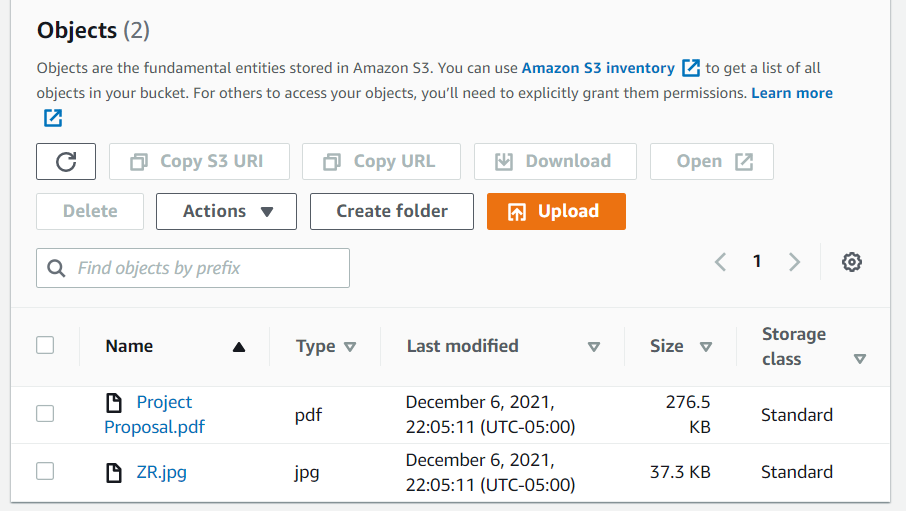
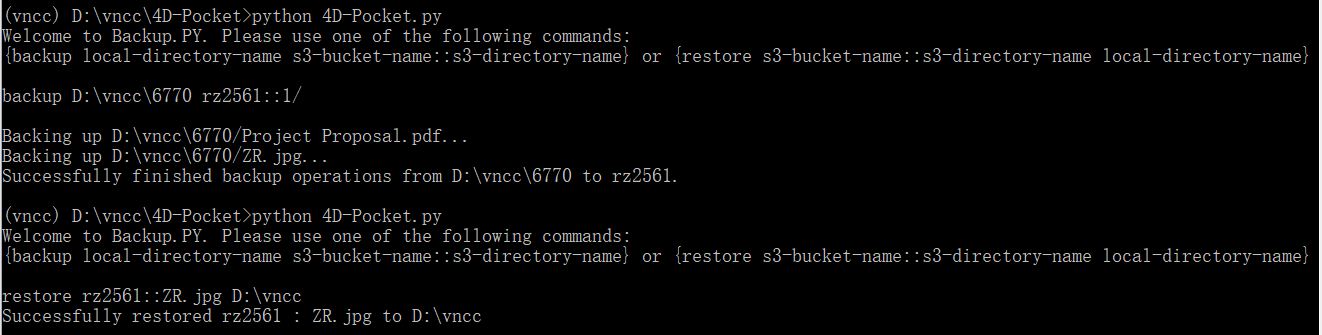
**7.Example**

You can start the program by running 4D-Pocket.py through your IDE or using command line.

Examples:

backup D:\vncc\6770 rz2561::1/

restore rz2561::ZR.jpg D:\vncc



**8.Code:**

Github : https://github.com/RayZhang-29/4D-Pocket