

Lab 3 Notes

Student ID: 22994257

Name: Gaoyuan Zhang

[Step 1] Preparation

1. Create the directories and files

```
mjieli@mjieli-VirtualBox:~$ mkdir rootdir
mjieli@mjieli-VirtualBox:~$ cd rootdir
mjieli@mjieli-VirtualBox:~/rootdir$ vim rootfile.txt
mjieli@mjieli-VirtualBox:~/rootdir$ vi rootfile.txt
mjieli@mjieli-VirtualBox:~/rootdir$ mkdir subdir
mjieli@mjieli-VirtualBox:~/rootdir$ cd subdir
mjieli@mjieli-VirtualBox:~/rootdir/subdir$ vim subfile.txt
mjieli@mjieli-VirtualBox:~/rootdir/subdir$ ls
subfile.txt
```

[Step 2] Save to S3

1. Download the `cloudstorage.py` from github and add some key functions, such as `create_bucket`, `upload_file`

```
# Given a root local directory, will return files in each level and
# copy to same path on S3
#
# -----

ROOT_DIR = '.'
ROOT_S3_DIR = '22994257-cloudstorage'

s3 = boto3.client("s3")
bucket_name = '22994257-cloudstorage'
bucket_config = {'LocationConstraint': 'ap-southeast-2'}

def upload_file(folder_name, file, file_name):
    response = s3.upload_file(file, bucket_name, file)
    print("Uploading %s" % file)

# Main program
# Insert code to create bucket if not there

try:
    response = s3.create_bucket(Bucket=bucket_name, CreateBucketConfiguration=bucket_config)
    print(response)
except Exception as error:
    pass

# parse directory and upload files

for dir_name, subdir_list, file_list in os.walk(ROOT_DIR, topdown=True):
    if dir_name != ROOT_DIR:
        for fname in file_list:
            upload_file("%s/" % dir_name[2:], "%s/%s" % (dir_name, fname), fname)

print("done")
```

2. Run the python code. It can create bucket, parse directory and upload files.

```
mjieli@mjieli-VirtualBox:~/lab3$ python3 cloudstorage.py
{'ResponseMetadata': {'RequestId': '12B3AGK454DDXA7D', 'HostId': 'JuLwh7+G0EfLzWSnZ/uCyFWxIyDAPyz60z0+3mRK3RtkTuxT5E53hbAGVya3kU4sXJKW/D2Z2LA=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'JuLwh7+G0EfLzWSnZ/uCyFWxIyDAPyz60z0+3mRK3RtkTuxT5E53hbAGVya3kU4sXJKW/D2Z2LA=', 'x-amz-request-id': '12B3AGK454DDXA7D', 'date': 'Mon, 15 Aug 2022 17:35:34 GMT', 'location': 'http://22994257-cloudstorage.s3.amazonaws.com/', 'server': 'AmazonS3', 'content-length': '0'}, 'RetryAttempts': 0}, 'Location': 'http://22994257-cloudstorage.s3.amazonaws.com/'}
Uploading ./rootdir/rootfile.txt
Uploading ./rootdir/subdir/subfile.txt
done
```

3. Confirm the result on S3 console. It has a right folder-file hierarchy.

Amazon S3 > Buckets > 22994257-cloudstorage > ./ > rootdir/

rootdir/

Objects | Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For other permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	rootfile.txt	txt	August 16, 2022, 01:35:35 (UTC+08:00)
<input type="checkbox"/>	subdir/	Folder	-

Amazon S3 > Buckets > 22994257-cloudstorage > ./ > rootdir/ > subdir/

subdir/

Objects | Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For other permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	subfile.txt	txt	August 16, 2022, 01:35:35 (UTC+08:00)

[Step 3] Restore from S3

1. Create a new program called `restorefromcloud.py`

```
mjieli@mjieli-VirtualBox:~/lab3$ vim restorefromcloud.py
```

2. Define a function `download_dir` to parse directory and download files. It will download the files from my bucket and put it in the `'/tmp'` directory.

```
import boto3
import os

def download_dir(client, resource, dist, local='/tmp', bucket='22994257-cloudstorage'):
    paginator = client.get_paginator('list_objects')
    for result in paginator.paginate(Bucket=bucket, Delimiter='/', Prefix=dist):
        if result.get('CommonPrefixes') is not None:
            for subdir in result.get('CommonPrefixes'):
                download_dir(client, resource, subdir.get('Prefix'), local, bucket)
        for file in result.get('Contents', []):
            dest_pathname = os.path.join(local, file.get('Key'))
            if not os.path.exists(os.path.dirname(dest_pathname)):
                os.makedirs(os.path.dirname(dest_pathname))
            if not file.get('Key').endswith('/'):
                resource.meta.client.download_file(bucket, file.get('Key'), dest_pathname)

client = boto3.client('s3')
resource = boto3.resource('s3')
download_dir(client, resource, './', '/tmp')
```

3. The `'/tmp'` directory before run the program

```
mjieli@mjieli-VirtualBox:/tmp$ ls
config-err-a49rTG
ssh-nI8JXoCX4qgU
systemd-private-b7c6fa972bba4b9487f63363ece3c887-colord.service-zKFqji
systemd-private-b7c6fa972bba4b9487f63363ece3c887-ModemManager.service-oMjuWg
systemd-private-b7c6fa972bba4b9487f63363ece3c887-switcheroo-control.service-JQ0IAf
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-logind.service-MgCpvi
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-resolved.service-0f0XGh
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-timesyncd.service-0Tp00e
systemd-private-b7c6fa972bba4b9487f63363ece3c887-upower.service-50io5e
tracker-extract-files.1000
tracker-extract-files.125
```

4. Run the program

```
mjieli@mjieli-VirtualBox:~/lab3$ python3 restorefromcloud.py
```

5. The `'/tmp'` directory after run the program

```
mjieli@mjieli-VirtualBox:~/lab3$ cd /tmp
mjieli@mjieli-VirtualBox:/tmp$ ls
config-err-a49rTG
rootdir
ssh-nI8JXoCX4qgU
systemd-private-b7c6fa972bba4b9487f63363ece3c887-colord.service-zKFqji
systemd-private-b7c6fa972bba4b9487f63363ece3c887-ModemManager.service-oMjuWg
systemd-private-b7c6fa972bba4b9487f63363ece3c887-switcheroo-control.service-JQ0IAf
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-logind.service-MgCpvi
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-resolved.service-0f0XGh
systemd-private-b7c6fa972bba4b9487f63363ece3c887-systemd-timesyncd.service-0Tp00e
systemd-private-b7c6fa972bba4b9487f63363ece3c887-upower.service-50io5e
tracker-extract-files.1000
tracker-extract-files.125
```

6. It has a right folder-file hierarchy.


```

mjieli@mjieli-VirtualBox:/tmp$ cd rootdir
mjieli@mjieli-VirtualBox:/tmp/rootdir$ ls
rootfile.txt  subdir
mjieli@mjieli-VirtualBox:/tmp/rootdir$ cd subdir
mjieli@mjieli-VirtualBox:/tmp/rootdir/subdir$ ls
subfile.txt

```

[Step 4] Write information about files to DynamoDB

1. Install jre and check the version

```

mjieli@mjieli-VirtualBox:~$ mkdir dynamodb
mjieli@mjieli-VirtualBox:~$ cd dynamodb
mjieli@mjieli-VirtualBox:~/dynamodb$ sudo apt-get install default-jre
[sudo] password for mjieli:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre openjdk-11-jre-headless
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni openjdk-11-jre openjdk-11-jre-headless
0 to upgrade, 9 to newly install, 0 to remove and 0 not to upgrade.
Need to get 39.6 MB of archives.
After this operation, 180 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://au.archive.ubuntu.com/ubuntu focal/main amd64 java-common all 0.72 [6,816 B]
Ign:2 http://au.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jre-headless
amd64 11.0.15+0ubuntu0.20.04.1
Get:3 http://au.archive.ubuntu.com/ubuntu focal/main amd64 default-jre-headless amd64 2:1.1
1-72 [3,192 B]
Get:4 http://au.archive.ubuntu.com/ubuntu focal/main amd64 ca-certificates-java all 2019040
5ubuntu1 [12.2 kB]
Ign:5 http://au.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jre amd64 11.
0.15+0ubuntu0.20.04.1
mjieli@mjieli-VirtualBox:~$ java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)

```

2. Get the file dynamodb_local_latest.tar.gz and unzip it.

```

mjieli@mjieli-VirtualBox:~/dynamodb$ ls
dynamodb_local_latest.tar.gz

mjieli@mjieli-VirtualBox:~/dynamodb$ tar -zxvf dynamodb_local_latest.tar.gz
DynamoDBLocal_lib/
DynamoDBLocal.jar

mjieli@mjieli-VirtualBox:~/dynamodb$ ls
DynamoDBLocal.jar      DynamoDBLocal_lib  README.txt
dynamodb_local_latest.tar.gz  LICENSE.txt        THIRD-PARTY-LICENSES.txt

```

3. Run the DynamoDB

```
mjieli@mjieli-VirtualBox:~/dynamodb$ java -Djava.library.path=./DynamoDBLocal_lib -jar Dyna
modBLocal.jar -sharedDb
Initializing DynamoDB Local with the following configuration:
Port:      8000
InMemory:   false
DbPath: null
SharedDb:   false
shouldDelayTransientStatuses: false
CorsParams: *
```

4. Run the `create_table.py`, which is used to create the table.

```
mjieli@mjieli-VirtualBox:~/dynamodb$ vim create_table.py
mjieli@mjieli-VirtualBox:~/dynamodb$ python3 create_table.py
dynamodb.Table(name='22994257-Table')
```

```
import boto3

dynamodb = boto3.resource('dynamodb')

table = dynamodb.create_table (
    TableName = '22994257-Table',
    KeySchema = [
        {
            'AttributeName': 'userId',
            'KeyType': 'HASH'
        },
        {
            'AttributeName': 'fileName',
            'KeyType': 'RANGE'
        }
    ],
    AttributeDefinitions = [
        {
            'AttributeName': 'userId',
            'AttributeType': 'S'
        },
        {
            'AttributeName': 'fileName',
            'AttributeType': 'S'
        }
    ],
    ProvisionedThroughput={
        'ReadCapacityUnits':1,
        'WriteCapacityUnits':1
    }
)
print(table)
```

5. Here is the key code. It can get some information about every file that is stored in S3 by reading the object's variable `cur` and `acl`, then write it to the DynamoDB table.

```
mjieli@mjieli-VirtualBox: ~/dynamodb x mjieli@mjieli-VirtualBox: ~/dynamodb x
import os
import boto3

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('22994257-Table')

BUCKET = '22994257-cloudstorage'
s3 = boto3.client('s3')
paginator = s3.get_paginator('list_objects_v2')

# List all of the objects
for page in paginator.paginate(Bucket=BUCKET):
    for cur in page.get("Contents", []):
        acl = s3.get_object_acl(Bucket=BUCKET, Key=cur['Key'])
        path, fileName = os.path.split(cur['Key'])
        lastUpdated = cur['LastModified'].strftime("%Y-%m-%d %H:%M:%S")
        userId = acl['Owner']['ID']
        owner = acl['Owner']['DisplayName']
        permissions = acl['Grants'][0]['Permission']
        response = table.put_item(Item={"userId": userId, "fileName": fileName, "path": path,
            "lastUpdated": lastUpdated, "owner": owner, "permissions": permissions })
        print(response)

mjieli@mjieli-VirtualBox:~/dynamodb$ python3 get_file_info.py
{'ResponseMetadata': {'RequestId': '7NRB06BANLC1M1QF7LONKTUVBFVV4KQNS05AEMVJF66Q9ASUAAJG',
'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Tue, 16 Aug 2022 16:38:04 GMT', 'content-type': 'application/x-amz-json-1.0', 'content-length': '2', 'connection': 'keep-alive', 'x-amzn-requestid': '7NRB06BANLC1M1QF7LONKTUVBFVV4KQNS05AEMVJF66Q9ASUAAJG', 'x-amz-crc32': '2745614147'}, 'RetryAttempts': 0}}
{'ResponseMetadata': {'RequestId': 'R143RCQB87T622CHUJENJCLR13VV4KQNS05AEMVJF66Q9ASUAAJG',
'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Tue, 16 Aug 2022 16:38:04 GMT', 'content-type': 'application/x-amz-json-1.0', 'content-length': '2', 'connection': 'keep-alive', 'x-amzn-requestid': 'R143RCQB87T622CHUJENJCLR13VV4KQNS05AEMVJF66Q9ASUAAJG', 'x-amz-crc32': '2745614147'}, 'RetryAttempts': 0}}
```

6. Finally, scan the table to check the result.

```
import boto3

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('22994257-Table')

response = table.scan()
print(response['Items'])

mjieli@mjieli-VirtualBox:~/dynamodb$ vim scan.py
mjieli@mjieli-VirtualBox:~/dynamodb$ python3 scan.py
[{'lastUpdated': '2022-08-15 17:35:35', 'fileName': 'rootfile.txt', 'path': './rootdir', 'userId': 'e899a06030e20f8e9945922db62a14112f50d6a6d19721ca733875beb9e50f3c', 'permissions': 'FULL_CONTROL', 'owner': 'mdanwarulkaikum.patwary'}, {'lastUpdated': '2022-08-15 17:35:35', 'fileName': 'subfile.txt', 'path': './rootdir/subdir', 'userId': 'e899a06030e20f8e9945922db62a14112f50d6a6d19721ca733875beb9e50f3c', 'permissions': 'FULL_CONTROL', 'owner': 'mdanwarulkaikum.patwary'}]
```

```
[{'lastUpdated': '2022-08-15 17:35:35',
'fileName': 'rootfile.txt',
'path': './rootdir',
'userId': 'e899a06030e20f8e9945922db62a14112f50d6a6d19721ca733875beb9e50f3c',
'permissions': 'FULL_CONTROL',
'owner': 'mdanwarulkaikum.patwary'},
{'lastUpdated': '2022-08-15 17:35:35',
'fileName': 'subfile.txt',
'path': './rootdir/subdir',
```

```
'userId': 'e899a06030e20f8e9945922db62a14112f50d6a6d19721ca733875beb9e50f3c',  
'permissions': 'FULL_CONTROL',  
'owner': 'mdanwarulkaium.patwary'}]
```