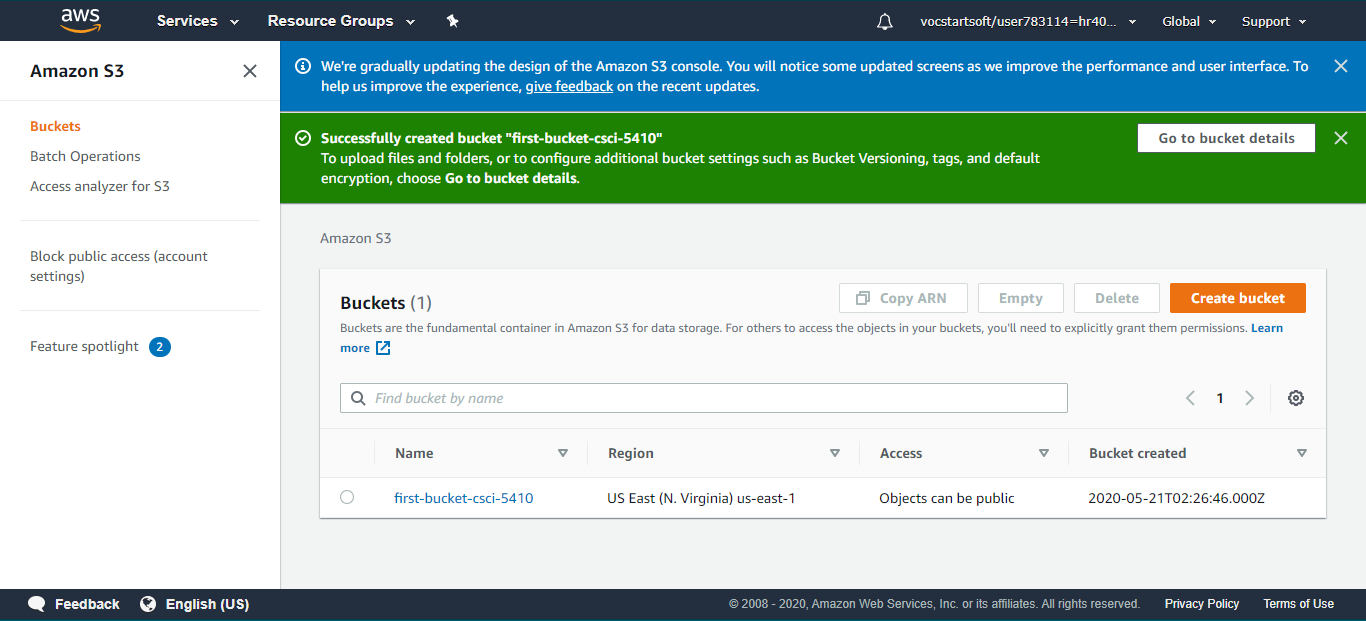
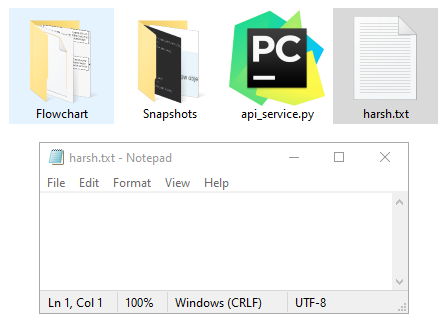
**Part B: AWS S3 storage experiment**

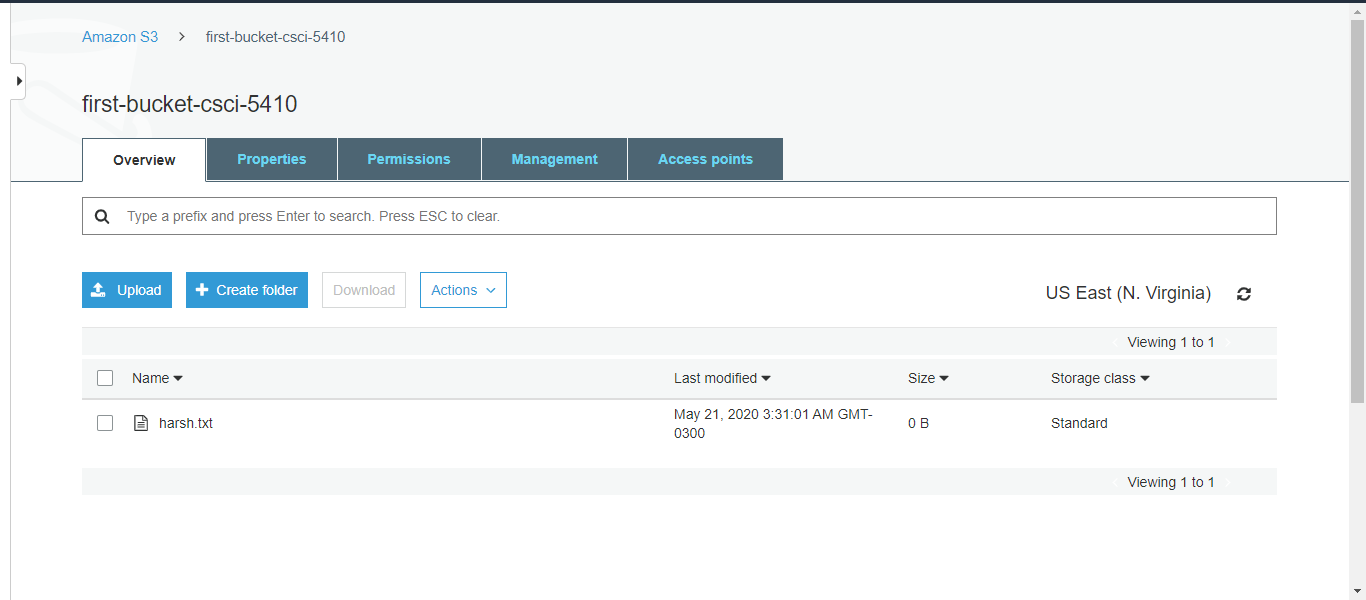
* 1. Create an empty bucket named “first-bucket-csci-5410”

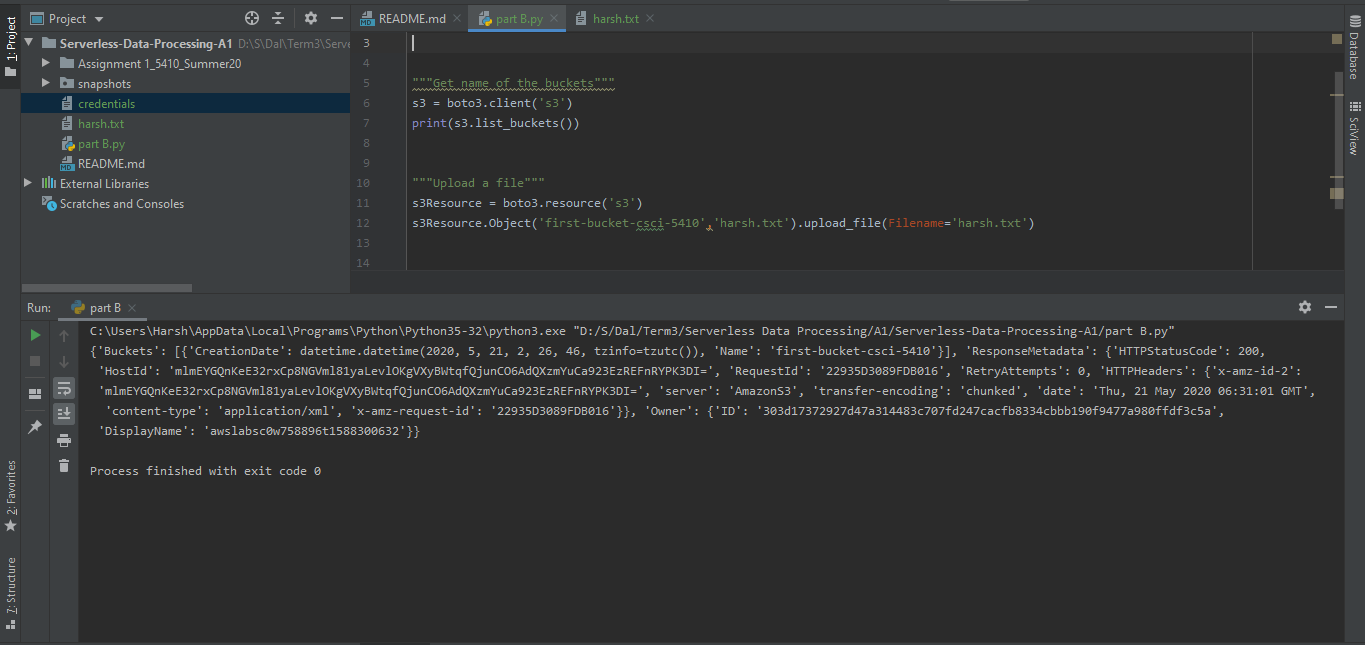


* 1. Create an empty text file ‘harsh.txt’

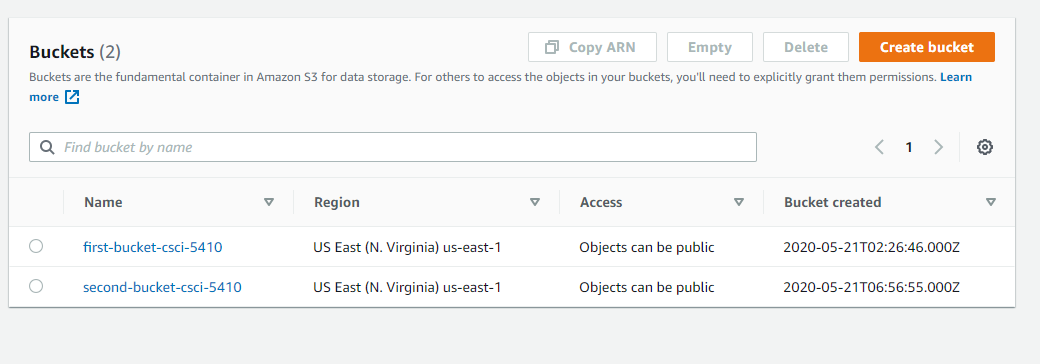


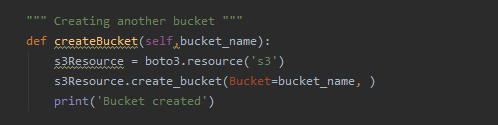
1. Upload the file on the bucket “first-bucket-csci-5410”

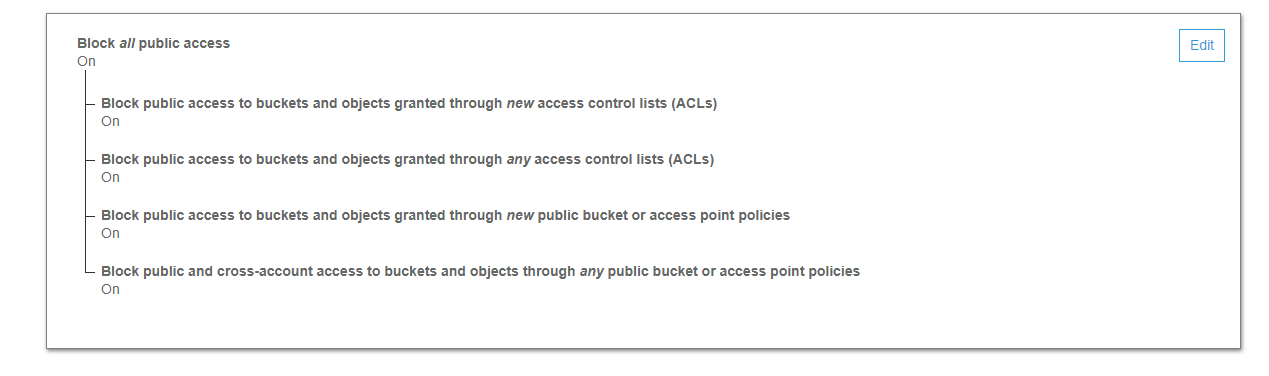


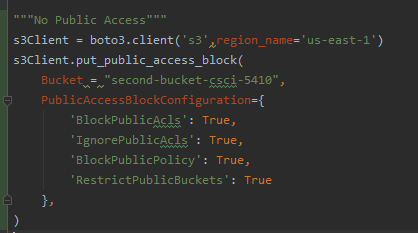


* 1. Create a second bucket named “second-bucket-csci-5410” using python.

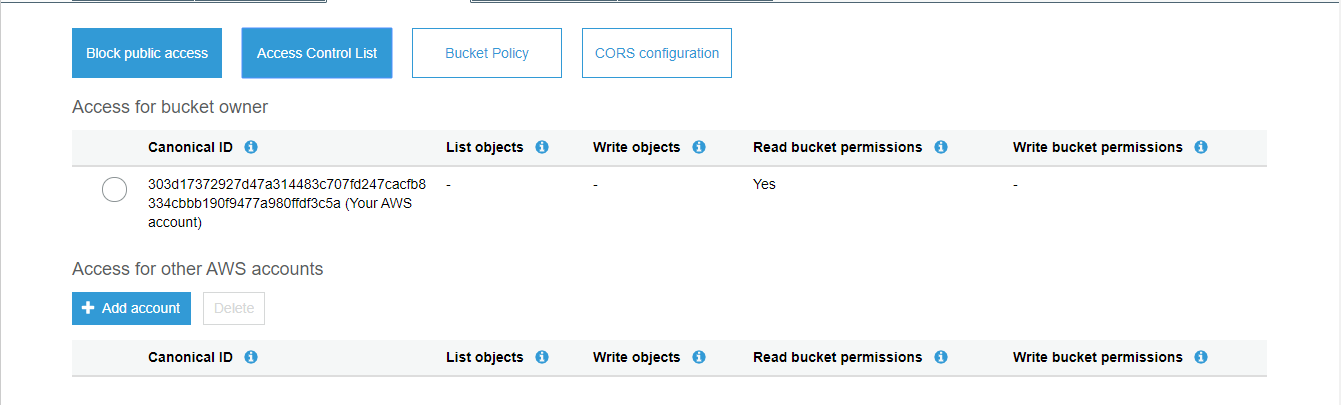


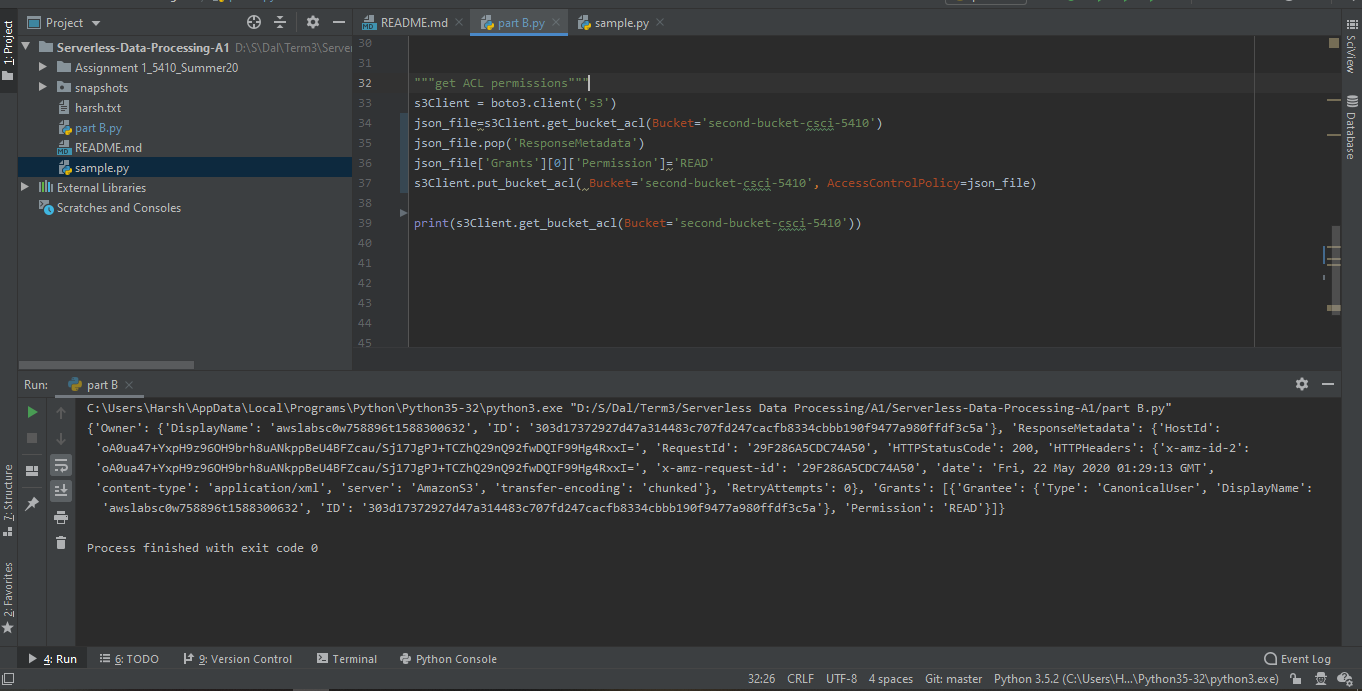


* 1. Change the access permission to “disable public access”

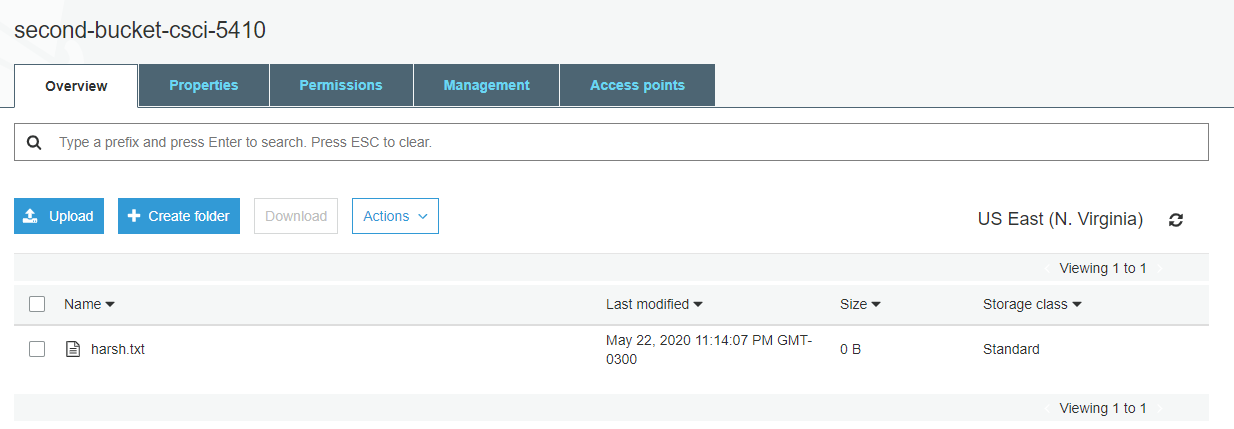


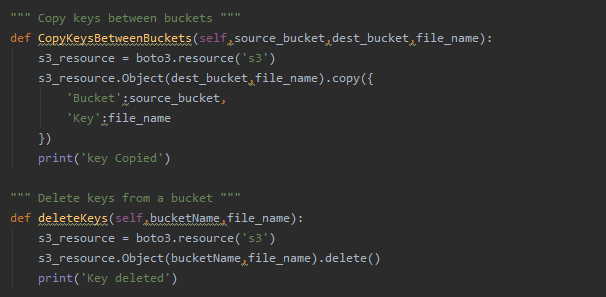
* 1. Change the ACL write option to “no” for bucket owner.



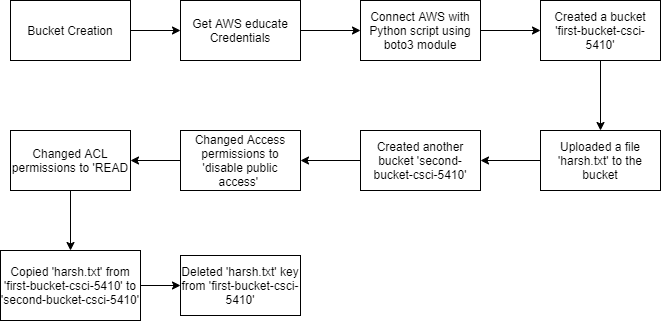


1. Move the file from “first-bucket-csci-5410” to “second-bucket-csci-5410”.





1. A flowchart showing the steps performed in the experiment



In the experiment, first and foremost I created a bucket 'first-bucket-csci-5410'. Then I started creating a user in IAM, but later I found that we cannot create IAM users in AWS educate account. Then I integrated my python script with AWS S3 using the credentials put in 'Users/<username>/'.aws/credentials file. After checking that my set up is complete, I uploaded 'harsh.txt' key to the bucket. Then I created another bucket named 'second-bucket-csci-5410' and changed its access permission using the 'put\_public\_access\_block()' method. I also changed its ACL permissions to 'READ' using 'put\_bucket\_acl()' method. Then I copied the key from former bucket to later created bucket and deleted the key from former bucket which ultimately moved the key.

**Source Code**

import boto3  
  
  
class s3Api:  
 *""" Get name of the buckets """* def listBuckets(self):  
 s3 = boto3.client('s3')  
 return s3.list\_buckets()  
  
 """ Upload a file """  
  
 def fileUpload(self, bucket\_name, source\_file\_name, file\_name):  
 s3Resource = boto3.resource('s3')  
 s3Resource.Object(bucket\_name, source\_file\_name).upload\_file(Filename=file\_name)  
 print('file uploaded')  
  
 """ Creating another bucket """  
  
 def createBucket(self, bucket\_name):  
 s3Resource = boto3.resource('s3')  
 s3Resource.create\_bucket(Bucket=bucket\_name, )  
 print('Bucket created')  
  
 """ Change bucket access (private, public etc) """  
  
 def manageBucketAccess(self, bucket\_name, block\_public\_acls=True, ignore\_public\_acls=True, block\_public\_policy=True, restrict\_public\_buckets=True, region='us-east-1'):  
 s3\_client = boto3.client('s3', region\_name=region)  
 s3\_client.put\_public\_access\_block(  
 Bucket=bucket\_name,  
 PublicAccessBlockConfiguration={  
 'BlockPublicAcls': block\_public\_acls,  
 'IgnorePublicAcls': ignore\_public\_acls,  
 'BlockPublicPolicy': block\_public\_policy,  
 'RestrictPublicBuckets': restrict\_public\_buckets  
 },  
 )  
 print('Access Changed')  
  
 """ Get ACL permissions """  
  
 def getAclPermissions(self, bucket\_name):  
 s3\_client = boto3.client('s3')  
 return s3\_client.get\_bucket\_acl(Bucket=bucket\_name)

""" Set ACL permissions """  
  
 def changeAclPermissions(self, bucket\_name, permission):  
 s3\_client = boto3.client('s3')  
 json\_file = s3\_client.get\_bucket\_acl(Bucket=bucket\_name)  
 json\_file.pop('ResponseMetadata')  
 json\_file['Grants'][0]['Permission'] = permission  
 s3\_client.put\_bucket\_acl(Bucket=bucket\_name, AccessControlPolicy=json\_file)  
 print('ACL permission changed')  
  
 """ Copy keys between buckets """  
  
 def copyKeysBetweenBuckets(self, source\_bucket, dest\_bucket, file\_name):  
 s3\_resource = boto3.resource('s3')  
 s3\_resource.Object(dest\_bucket, file\_name).copy({  
 'Bucket': source\_bucket,  
 'Key': file\_name  
 })  
 print('key Copied')  
  
 """ Delete keys from a bucket """  
  
 def deleteKeys(self, bucketName, file\_name):  
 s3\_resource = boto3.resource('s3')  
 s3\_resource.Object(bucketName, file\_name).delete()  
 print('Key deleted')  
  
 """ Downloading a file from the bucket """  
  
 def downloadFile(self, bucket\_name, obj\_name, dest\_file\_name):  
 s3Client = boto3.client('s3')  
 s3Client.download\_file(bucket\_name, obj\_name, dest\_file\_name)  
 print('file downloaded')  
  
  
s3 = s3Api()  
s3.listBuckets()  
s3.fileUpload("first-bucket-csci-5410", "harsh.txt", "harsh.txt")  
s3.createBucket("second-bucket-csci-5410")  
s3.manageBucketAccess("second-bucket-csci-5410")  
s3.getAclPermissions("second-bucket-csci-5410")  
s3.changeAclPermissions("second-bucket-csci-5410", "READ")  
s3.copyKeysBetweenBuckets("first-bucket-csci-5410", "second-bucket-csci-5410", "harsh.txt")  
s3.deleteKeys("first-bucket-csci-5410", "harsh.txt")  
s3.downloadFile("first-bucket-csci-5410", "Lookup5410.txt", "Lookup.txt")

**References**

[1]"Python S3 Examples — Ceph Documentation", *Docs.ceph.com*, 2020. [Online]. Available: https://docs.ceph.com/docs/master/radosgw/s3/python/. [Accessed: 25- May- 2020].

[2]"Amazon S3 examples — Boto3 Docs 1.13.16 documentation", *Boto3.amazonaws.com*, 2020. [Online]. Available: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/s3-examples.html. [Accessed: 25- May- 2020].

[3]R. Python, "Python, Boto3, and AWS S3: Demystified – Real Python", *Realpython.com*, 2020. [Online]. Available: https://realpython.com/python-boto3-aws-s3/. [Accessed: 25- May- 2020].