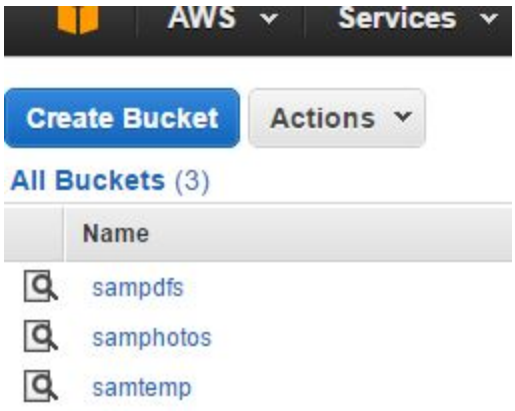
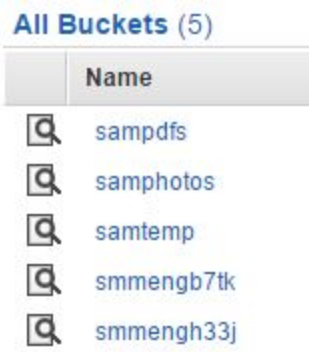
Step 1. Create at least 2 buckets in S3

Before Step 1



```
>>> UPLOAD_PATH = '/home/siming.meng/uploads/'
>>> DOWNLOAD_PATH = '/home/siming.meng/Downloads/'
>>> BUCKET_PREFIX = 'smmeng'
>>>
>>> s3 = boto.connect_s3()
>>> rs = s3.get_all_buckets()
>>>
>>> print 'Listing all buckets'
Listing all buckets
>>> for b in rs:
...     print b
...
<Bucket: sampdfs>
<Bucket: samphotos>
<Bucket: samtemp>
>>> newBucketsToCreate = raw_input("How many new bucket to create
How many new bucket to create (less than 3):2
>>> newBucketList = []
>>> for num in range(0, int(newBucketsToCreate)):
...     newBucketName = BUCKET_PREFIX + random_generator().lower()
...     print 'newBucketName=[', newBucketName
...     nb=s3.create_bucket(newBucketName)
...     newBucketList.append( newBucketName)
...
newBucketName=[ smmengb7tk
newBucketName=[ smmengh33j
>>> print 'relisting all buckets again'
relisting all buckets again
>>> rs = s3.get_all_buckets()
>>> for b in rs:
...     print b
...
<Bucket: sampdfs>
<Bucket: samphotos>
<Bucket: samtemp>
<Bucket: smmengb7tk>
<Bucket: smmengh33j>
>>>
```

After step 1



Step 2. Get filenames from the user for existing files

– store these files on objects in the buckets you create

List the content before uploading as paths are preset

Uploading

```
iFileName =
iFileList = []iFileName = ''
>>> iFileList = []
>>> while True:
...     iFileName = raw_input("Enter local file name to upload (use 'EOF' t
...     if  iFileName == 'EOF' or len(iFileName)==0:
...             break;
...     iFileList.append(iFileName)
...
Enter local file name to upload (use 'EOF' to stop):Class1.pdf
Enter local file name to upload (use 'EOF' to stop):Class2.pdf
Enter local file name to upload (use 'EOF' to stop):Class3.pdf
Enter local file name to upload (use 'EOF' to stop):Class4.pdf
Enter local file name to upload (use 'EOF' to stop):Class5.pdf
Enter local file name to upload (use 'EOF' to stop):
>>> print 'iFileList=', iFileList
iFileList= ['Class1.pdf', 'Class2.pdf', 'Class3.pdf', 'Class4.pdf', 'Class5
>>>
>>> for bucketname in newBucketList:
...     for filename in iFileList:
...             bucket=s3.get_bucket(bucketname)
...             anothernewkey=bucket.new_key(filename)
...             anothernewkey.set_contents_from_filename(UPLOAD_PATH+filena
...             anothernewkey.set_acl('public-read')
...

761423
556788
143790
116238
391039
761423
556788
143790
116238
391039
>>>
```

Step 3. List all buckets and objects in the buckets

```
>>> for bucketname01 in newBucketList:
...     bucketlist=s3.get_bucket(bucketname01)
...     bucket=s3.get_bucket(bucketname01)
...     for obj in bucketlist:
...             print obj
...
<Key: smmengb7tk,Class1.pdf>
<Key: smmengb7tk,Class2.pdf>
<Key: smmengb7tk,Class3.pdf>
<Key: smmengb7tk,Class4.pdf>
<Key: smmengb7tk,Class5.pdf>
<Key: smmengh33j,Class1.pdf>
<Key: smmengh33j,Class2.pdf>
<Key: smmengh33j,Class3.pdf>
<Key: smmengh33j,Class4.pdf>
<Key: smmengh33j,Class5.pdf>
>>>
```

uckets / smmengh33j

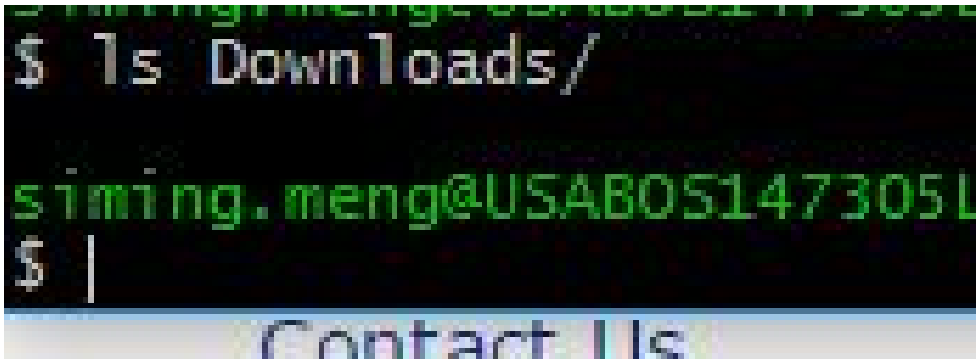
Name	Storage Class	Size	Last Modified
Class1.pdf	Standard	743.5 KB	Sun Sep 18 09:3
Class2.pdf	Standard	543.7 KB	Sun Sep 18 09:3
Class3.pdf	Standard	140.4 KB	Sun Sep 18 09:3
Class4.pdf	Standard	113.5 KB	Sun Sep 18 09:3
Class5.pdf	Standard	381.8 KB	Sun Sep 18 09:3

ıckets / smmengb7tk

Name	Storage Class	Size	Last Modified
Class1.pdf	Standard	743.5 KB	Sun Sep 18 09:3
Class2.pdf	Standard	543.7 KB	Sun Sep 18 09:3
Class3.pdf	Standard	140.4 KB	Sun Sep 18 09:3
Class4.pdf	Standard	113.5 KB	Sun Sep 18 09:3
Class5.pdf	Standard	381.8 KB	Sun Sep 18 09:3

Step 4. Download the objects stored in the Buckets

Before downloading



After recursively downloading

```
...     bucketlist=s3.get_bucket(bucketname01)
...     if os.path.exists(bucketname01)== False :
...                     print "creating folder [", bucketname01
...                     os.mkdir(DOWNLOAD_PATH+bucketname01)
...
...     for obj in bucketlist:
...             fn = bucketname01 + "/" + str(obj.key)
...             print obj, " filename=", fn, fn[len(fn)-1]
...             if fn[len(fn)-1] is '/' and os.path.exists(fn)== False
...                     print "creating folder [", fn
...                     os.mkdir(DOWNLOAD_PATH+ fn)
...             elif fn[len(fn)-1] is '/':
...                     continue
...             else:
...                     obj.get_contents_to_filename(DOWNLOAD_PATH+fn)
...
creating folder [ smmengb7tk
<Key: smmengb7tk,Class1.pdf>  filename= smmengb7tk/Class1.pdf f
<Key: smmengb7tk,Class2.pdf>  filename= smmengb7tk/Class2.pdf f
<Key: smmengb7tk,Class3.pdf>  filename= smmengb7tk/Class3.pdf f
<Key: smmengb7tk,Class4.pdf>  filename= smmengb7tk/Class4.pdf f
<Key: smmengb7tk,Class5.pdf>  filename= smmengb7tk/Class5.pdf f
creating folder [ smmengh33j
<Key: smmengh33j,Class1.pdf>  filename= smmengh33j/Class1.pdf f
<Key: smmengh33j,Class2.pdf>  filename= smmengh33j/Class2.pdf f
<Key: smmengh33j,Class3.pdf>  filename= smmengh33j/Class3.pdf f
<Key: smmengh33j,Class4.pdf>  filename= smmengh33j/Class4.pdf f
<Key: smmengh33j,Class5.pdf>  filename= smmengh33j/Class5.pdf f
>>> print "SUCCESS!"
SUCCESS!
```

```
$ !ls
ls Downloads/
smmengb7tk/  smmengh33j/

siming.meng@USABOS147305L ~
$ ls Downloads/*
Downloads/smmengb7tk:
Class1.pdf  Class2.pdf  Class3.pdf  Class4.pdf  Class5.pdf

Downloads/smmengh33j:
Class1.pdf  Class2.pdf  Class3.pdf  Class4.pdf  Class5.pdf
```