**ECS Demo Cookbook**

**Accessing ECS Objects Using Python (Windows 7)**

**Using both the S3 and SWIFT API**

1. Create an account on the ECS Test Drive site 🡪 <https://portal.ecstestdrive.com/>
   1. Once the account is created, make sure to create at least ONE secret key for S3! Click on CREDENTIALS 🡪 MANAGE SECRET KEYS 🡪 CREATE SECRET KEY
   2. Create a SWIFT Password! Click on CREDENTIALS 🡪 Swift Password 🡪 SET PASSWORD (make it something EASY to remember!)
   3. Click on CREDENTIALS once again, and keep the site open. You will need it to establish access with your programs and tools.
2. Install S3 Browser ([http://s3browser.com/)](http://s3browser.com/)%20) and/or Cyberduck (<https://cyberduck.io/>)
3. Enter credentials for desktop tool to establish connectivity.
   1. Cyberduck S3:
      1. Click CONNECTION.
      2. Select “S3 Amazon Simple Storage Service” from drop down list
      3. Cut and paste the END POINT (object.ecstestdrive.com) from the ECS Test Drive CREDENTIALS site into the SERVER field
      4. Cut and paste Access key and Secret Access Key into appropriate fields
   2. Cyberduck SWIFT:
      1. Click CONNECTION.
      2. Select “Swift Openstack Object Storage” from drop down list
      3. Cut and paste the END POINT (swift.ecstestdrive.com) from the ECS Test Drive CREDENTIALS site into the SERVER field
      4. Cut and paste Tenant ID, Secret Key and Password into appropriate fields
   3. S3 Browser:
      1. Select ACCOUNTS 🡪 ADD NEW ACCOUNT
      2. Choose any name
      3. Select “S3 Compatible Storage” from drop down list
      4. Cut and paste End point (object.ecstestdrive.com) and Keys from ECS Test Drive CREDENTIALS into appropriate fields.
4. Install Python. V2.7 release (<https://www.python.org/ftp/python/2.7.10/python-2.7.10.msi>)
   1. Run the msi to install.
   2. Add c:\python27\ to the PATH system environment variable (My Computer 🡪 Properties 🡪 Advanced System Settings 🡪 Environment Variables 🡪 Path. Then click edit and add the value to the end.
5. Install boto. Boto is the Python library used to access AWS S3 (AWS SDK).
   1. Cd c:\python27\scripts
   2. Pip install boto
6. Install the python SWIFT library
   1. Cd c:\python27\scripts
   2. Pip install python-swiftclient
7. Write your scripts! Or, install the four files I included in a new directory on your laptop.
   1. For S3 Demo: Run the ecsdemo.py script (c:\python.exe ecsdemo.py). I included lots of pauses to allow you to observe output using Cyberduck, S3 Browser, or Windows Explorer…depending on where the file/object is moved or created.
   2. For SWIFT Demo: Run the viprtest-swift.py script (c:\python.exe viprtest-swift.py). The code is heavily documented and includes lots of pauses.
8. Interesting Links:
   1. Good Samples at CEPH 🡪 <http://ceph.com/docs/master/radosgw/s3/python/>
   2. Intro to boto 🡪 <http://boto.readthedocs.org/en/latest/s3_tut.html#creating-a-connection>
   3. EMC Getting Started with Python and ECS Page 🡪 <https://community.emc.com/docs/DOC-37048>
   4. EMC Getting Started with ALL SDK’s 🡪 <https://community.emc.com/docs/DOC-27910>

**S3 Integration DEMO Script**

1. Make sure the customer knows the basics of using S3 with ECS storage
2. Open Python script using the IDLE editor to review the different sections. Each section of the demo is explained in the comments.
3. Open a Windows Explorer Screen (go to demo directory), Command Prompt (go to demo directory), the ECS Test Drive Website, and S3 Browser or Cyberduck . Strategically position so you can see all screens simultaneously. I use S3 Browser.
4. Walk through how to create an ECS Test Drive account and create a secret key. Then, open the CREDENTIALS page. This is needed for the next step.
5. Select ACCOUNTS from S3 Browser and explain how to establish a connection to ECS Test Drive
6. From S3 Browser, Note that there are no buckets currently created. Press the BUCKETS 🡪 REFRESH to confirm.
7. Using Windows Explorer, show the two text files in the DEMO directory:
   1. Frompc.txt is used in the USE S3 TO MAKE A LOCAL COPY OF A FILE
   2. Copyfile.txt is used to demonstrate a PUT from the local PC to the S3 bucket.
8. Explain that the demo was created to show some capabilities when using S3 object storage from within program code:
   1. Create a bucket
   2. List contents of a bucket
      1. After complete…show bucket in S3 browser (refresh)
   3. Create and add data to an object
      1. After complete…show object in S3 browser (refresh)
   4. Generate a PUBLIC URL for that object
   5. Delete an object
      1. After complete…show bucket in S3 browser (refresh)
   6. PUT an object from the local PC to the S3 bucket
      1. After complete…show object in S3 browser (refresh)
   7. Use S3 to make a copy of a local file with a new name
      1. While in progress…show object in S3 browser (refresh)
      2. Once complete, show object using Windows Explorer
   8. Delete the object and bucket
      1. After complete…show in S3 Browser
9. When complete, delete topc.txt file from demo directory.