Assignment 1: AWS S3 Storage Shell (S5)

Overview

This assignment will require that you acquire a deep knowledge of the AWS Boto3 SDK for Python. You will be designing and implementing a *shell* that allows an AWS user (who has proper credentials) to easily access and manipulate their S3 objects. The user experience should be comparable to the experience of using a local file system. In fact, your shell should allow the user to manipulate files on both their local system and the S3 system and between the two systems. The shell should present a seamless experience for the user - this shell will allow the user to manipulate their S3 resources easily and efficiently.

Notes

- You must use the Boto3 SDK to create your shell. You cannot just call the AWS CLI.
- A full S3 location description for an object is composed of two parts: the bucket name and the full pathname for the object. The format is

  <bucket name>:<pathname>

  As an example, if you have a bucket named *first4010bucket* and in this bucket you have an object that acts like a *folder* called *notes* and in that folder you have an object called *lecture1.docx* then the full S3 location description for the object that is a MSWord document is

  first4010bucket:notes/lecture1.docx
- You might notice that the pathname for this object starts with "/". "/" is used to indicate the S3 location that contains your buckets - think of it as the *root* of your S3 storage system.
- All of your shell's (**S5**) commands return 0 if successful and 1 if unsuccessful. If the command is a regular shell command that is to be executed by the user's session shell (see 3a below) then the return is whatever the session shell returns.
- To exit your **S5** shell you will type either *quit* or *exit*.

Requirements

1. AWS authentication will rely on the user having a configuation/authentication file in their current directory so that the shell can establish a connection to S3. The shell will use a file in the user's current directory (same directory as the S5 shell program) called *S5-S3conf* to establish a connection to AWS and S3. This should happen as soon as the shell is run and the message:

   Welcome to the AWS S3 Storage Shell (S5)
   You are now connected to your S3 storage

   should appear if an authorized connection to the user's S3 storage has been successfully established. Otherwise the message will be:

   Welcome to the AWS S3 Storage Shell (S5)
   You could not be connected to your S3 storage
   Please review procedures for authenticating your account on AWS S3

2. The prompt for the shell will be

   S5>

   and your location in your S3 space is at the *root* of your S3 space, *i.e.* you must either create a bucket and then *chfolder* to it to establish a location in S3 space or if you already have buckets then you must change to one of these buckets before you can do any Cloud commands other than *create _bucket* or *list* objects (see below).

3. Local File Functions

   a) Your shell should pass all non-Cloud related commands to your session's shell (bash, zsh, PowerShell, etc.). Thus, you can use commands like *ls* to view your local files.

   b) copy local file to Cloud location

Function	copies a local file to a Cloud (S3) location
Format	*lc_copy <full or relative pathname of local file> <bucket name>:<full pathname of S3 object>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* "Unsuccessful copy"
Examples	*S5> lc_copy catpictures/mycat01.jpg cis4010b01:images/cats/mycat.jpg*

c) copy Cloud object to local file system

Function	copies a Cloud object to a local file system location
Format	*cl_copy <bucket name>:<full pathname of S3 file> <full pathname of the local file>* *cl_copy <full or relative pathname of S3 file> <full pathname of the local file>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* "Unsuccessful copy"
Examples	*S5> cl_copy cis4010b01:images/cats/mycat.jpg catpic001.jpg* *S5> cl_copy mycat.jpg catpic001.jpg* (in this case the S3 object is in the current working directory and the file is in the local current working directory)

4. Cloud Functions

a) create bucket

Function	creates a bucket in the user's S3 space following naming conventions for S3 buckets
Format	*create_bucket <bucket name>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* " Cannot create bucket "
Examples	*S5> create_bucket cis4010b01*

b) create directory/folder

Function	creates a *directory* or *folder* in an existing S3 bucket
Format	*create_folder <bucket name>:<full pathname for the folder>* *create_folder <full or relative pathname for the folder>* (this implies that you have already used *ch_folder* to move to some location in your S3 space)
Returns	*on success*: command prompt *on failure*: error message, *e.g.* " Cannot create folder "
Examples	*S5> create_folder cis4010b01:images* *S5> create_folder cis4010b01:images/cats* *S5> ch_folder cis4010b01* *S5> create_folder images* *S5> ch_folder images* *S5> create_folder cats*

c) change directory/folder

Function	changes the current working folder/directory in your S3 space
Format	*ch_folder <bucket name>* *ch_folder <bucket name>:<full pathname of directory>* *ch_folder <full or relative pathname of directory>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* " Cannot change folder "
Examples	*S5> ch_folder cis4010b01* *S5> ch_folder cis4010b01:images/cats* *S5> ch_folder cis4010b01* *S5> ch_folder images/cats* *S5> ch_folder /* *S5> ch_folder ..*

	S5> ch_folder ../..

d) current working folder/directory

Function	displays the current working folder/directory, i.e. your location in S3 space. If you are not yet located in a bucket, then the response will be "/". This would be the response if you execute this command after starting the shell.
Format	*cwf*
Returns	*on success*: <bucket name>:<full pathname of directory> *on failure*: error message, *e.g.* "Cannot access location in S3 space"
Examples	*S5> cwf* */* *S5> ch_folder cis4010b01:images* *S5> cwf* *cis4010b01:images* *S5> ch_folder cats* *S5> cwf* *cis4010b01:/images/cats*

e) list buckets, folders, objects

Function	like the Unix ls command, list will show either a short or long form of the contents of your current working directory or a specified S3 location (including "/"). The argument to get the long version is "-l" - same as the Unix ls command
Format	*list* *list <bucket name>* *list <bucket name>:<full pathname for directory or file>*
Returns	*on success*: displays the S3 location's content similar to the Unix ls command *on failure*: error message, *e.g.* " Cannot list contents of this S3 location"
Examples	*S5> list cis4010b01* *S5> list -l cis4010b01:images/cats* *S5> list /* *S5> chdir cis4010b01* *S5> list -l*

f) copy objects

Function	copy an object from one S3 location to another
Format	*ccopy <from S3 location of object> <to S3 location>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* "Cannot perform copy"
Examples	*S5> ccopy cis4010b01:images/cats/pichappycat.png pic001.png* *S5> ccopy pic001.png cis4010b1:backups/pic001.png*

g) delete object

Function	delete an object (folders included but only if they are empty). This command will not delete buckets.
Format	*cdelete <full or indirect pathname of object>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* "Cannot perform delete"
Examples	*S5> cdelete cis4010b01:images/cats/pic001.png* *S5> cdelete pic001.png* *S5> cdelete cis4010b01:images/cats*

h) delete bucket

Function	delete a bucket if it is empty. You cannot delete the bucket that you are currently in.
Format	*delete_bucket <bucket name>*
Returns	*on success*: command prompt *on failure*: error message, *e.g.* "Cannot delete bucket"
Examples	*S5> delete_bucket cis4010b01*

Hints

- Shells are easy to write in Python. There are commands in the *os* and *sys* Python modules such as *os.system* that make running commands in your session shell extremely simple. Python also has lots of easy ways to parse the commands in your shell. Honestly the loop to run your shell is basically several lines of code. Communicating and managing S3 resources will take a bit of research into *boto3* but there are some very good tutorials and manuals online.

- You will be expanding this shell in Assignment 5 so think deeply about the design for this assignment. A good, clean design for this shell will help with making Assignment 5 a breeze!

- So, what error messages should S5 produce? Well, that is up to you but, you will be rewarded for helpful error messages.

- What is the running environment for the S5 program? Does it matter if you are running on Linux, Mac OSX, or Windows? I am going to leave it to you to think about this and ponder how you handle this portability issue.