Name: Thuan Tran
CSS 490 Lab 3

DESIGN DOC

Table of Contents

What is included ?

Within this submission, I have included a zip file which contains the following:

- This design doc
- A compiled Java .class file that can be run from the command line
- Other folders that contain the dependencies for the project
- A Lab3.zip file that contains the source code and pom file to use for Maven

How to run this ?

Important: This requires Java 8

From the same folder where you extracted the files with the .class file, run:

java Main <absolute path to the place where you want to back up>

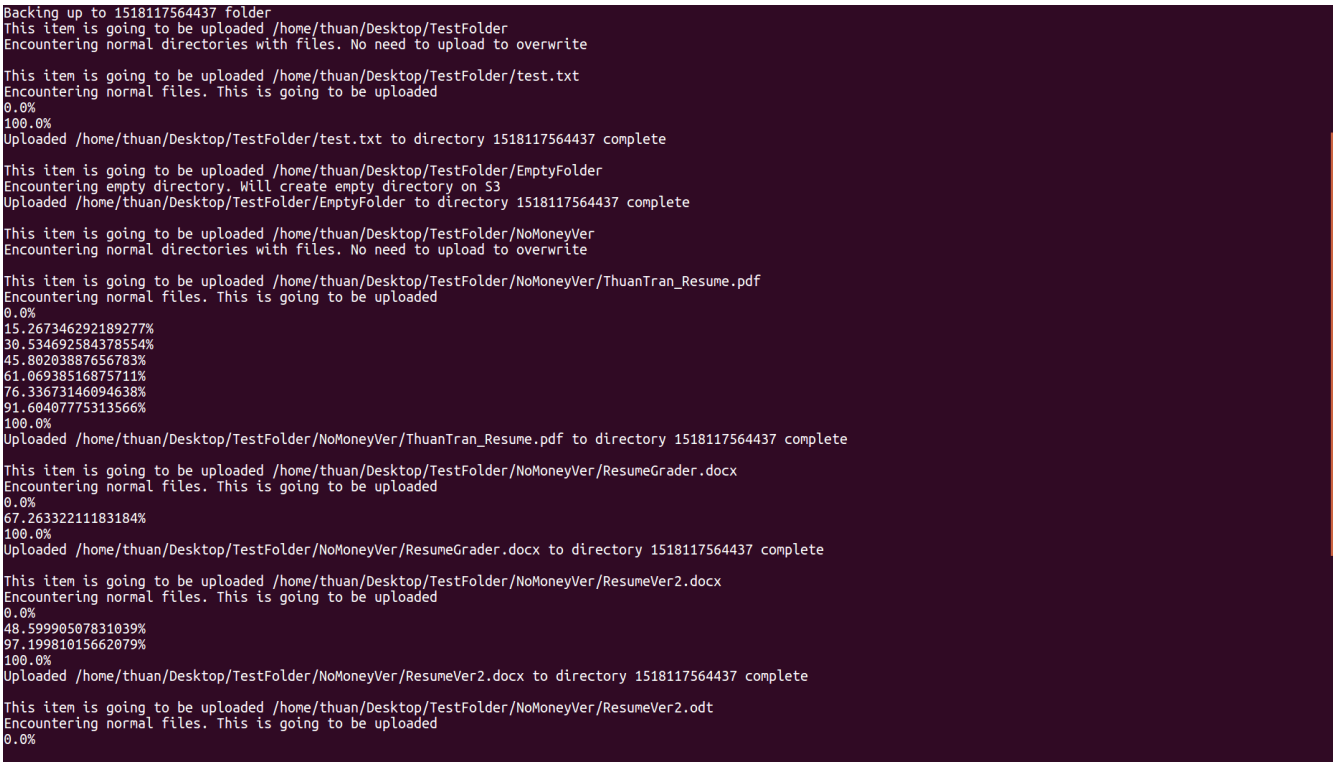
For example:

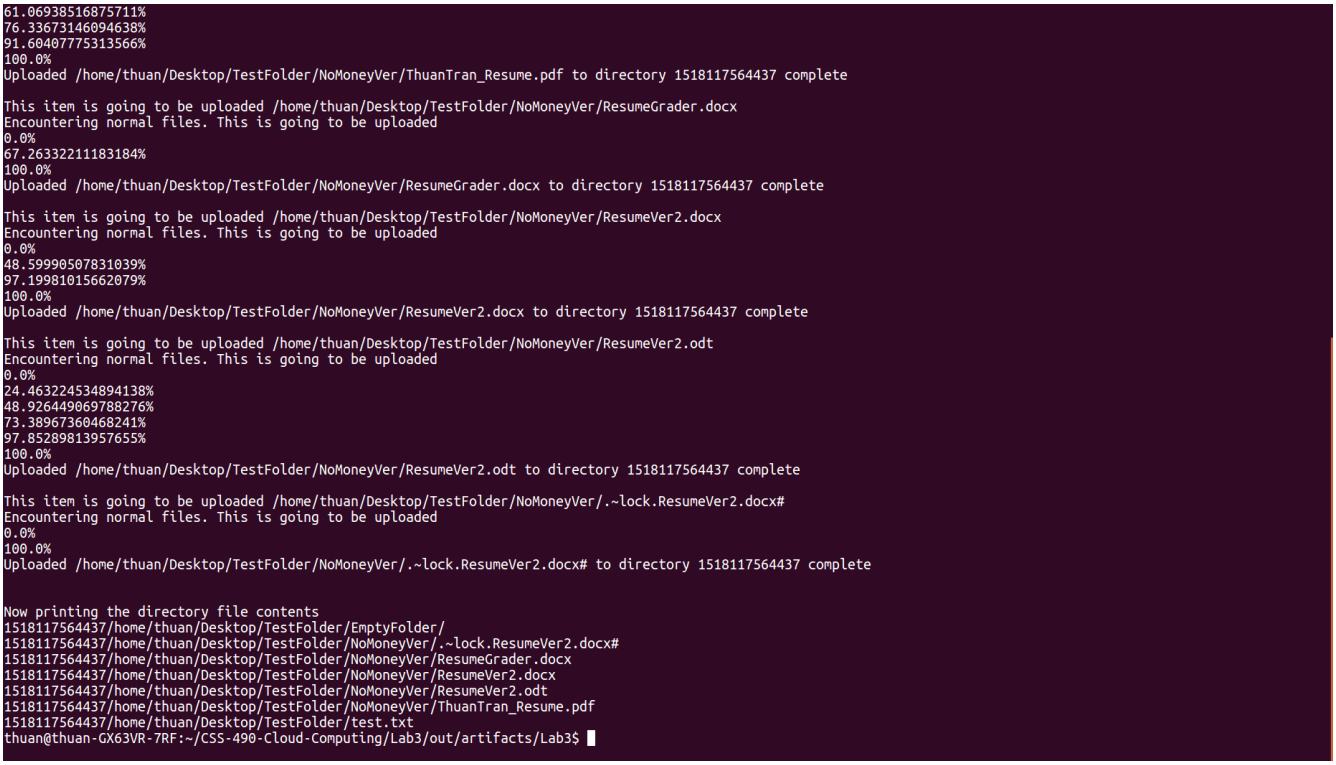
java Main /home/thuan/Desktop/TestFolder (Be careful with the path: /home vs home are different)

The program will upload each individual item one by one into a unique folder on S3 to avoid overwrite. In addition, printing out the percentage plus querying the data to make sure the data got uploaded

* Note: Assuming your machine have aws cli and does not have internet restrict (I tested this on the school's linux lab but since aws cli was not installed + somehow they block the traffic to AWS, it does not run. It does run on normal linux machines).

** Just to be caution as well, you can open the source code and look for the access key and secret key to configure on the command line using "aws configure". Selecting the region as "us-west-2" and format as "json" . Note that this step is not required but like a last resource if you can't run this





Architectural design and logic flow

Since this is a straight forward program, I decided a simple command line invoke would be enough.
The program has 4 parts:

- Main method: The main entry point
- Helper method to set up S3 and Transfer manager Client
- Helper method to decide what type of upload to use
- Helper method to get the info on S3 after backup to verify

When the user enter a path to the backup location, the program will utilize depth first search (Stream) to get all the folders underneath it
It will continue to set up Amazon S3 and Transfer Manager to begin the upload.
For each file that it will upload, the program will determine what kind of upload it will use. There are 3 types of upload that we can encounter:

- It the file is a normal file: upload it
- If the file is an empty directory: create empty content and upload on S3 with a “/”
- For remaining cases, skip because when we upload a file, we already have created the directory on S3

While the upload is in progress, print out the percentage
When all files have been uploaded, connect to S3 and print out the content of the files we just uploaded

Design decisions

How to provide access ?

One of the first design decision that I have to solve is: how to let you (the grader) run this on your machine. After a while, I decided to create a separate account using AWS IAM (Restricted access to S3) and hard-coded the credentials inside the source file so that the program can run directly everywhere

Why did I use a unique directory number each time ?

Another design decision is how to keep the backup on S3. If I just put them there without any separation, then it will be come a mess in no time. So I decided to employ the backup strategy of Linux where each backup has a key that uniquely identify it. In this case, the directory number generated from the system.

A little bit extra

Maybe you want to use the GUI instead ? Well, I have just the one for you. I have created a user account that you can use to browse S3 to see the files

Account alias :	css490
ID:	lab3user
Password:	css490lab3

Remainder

Please don’t share the above account info, the secret and access key

In addition, while I am testing my program, I have exceeded the free -tier already. So now, every call to S3 will have an impact on me. Please use it appropriately.

In addition, the files that was backup to the S3 have private access. So you can’t open a link directly on the browser to it. You will have to download it to open it.