**Data Uploader to Redshift Cluster**

This project contains a program that uploads files to the Amazon Redshift Cluster. This program allows the user to upload one or multiple files to the cluster and stores each file as a table in the database. This is useful for those who want to automate uploading files to redshift.

The process

Note that in order to use this project, you need to create a redshift cluster first.

**Installation**

To install this program, clone the repository.

**Dependencies**

This program depends on:

* Python 2.7.x or higher
* pip
* boto3
* psycopg2
* requests

If you do not have pip installed on your platform, follow the installation instructions [here](https://pip.pypa.io/en/latest/installing.html#install-or-upgrade-pip). Alternatively, you can install [Anaconda](https://www.continuum.io/downloads).

To install boto3:

Sudo pip install boto3

To install boto3:

pip install psycopg2

To install requests:

pip install requests

**Optional**

Although the program has an option for entering your access id, and secret id of your aws console, the following can be an additional step that can be followed to save your aws credentials. As a result of this, you do not need need enter your access key and secret id in the program itself.

**AWS Credentials**

To be able to connect to your AWS account, you need to add your AWS credentials.

1. Go to [AWS IAM Console](https://console.aws.amazon.com/iam/home).
2. Create a new user or use an existing user.
3. Generate a new set of keys for the user.
4. Create a file named credentials in the directory ~/.aws/ (Create ~/.aws/ if it's not already there).
5. Put the following in credentials file:

aws\_access\_key\_id = YOUR\_ACCESS\_KEY

aws\_secret\_access\_key = YOUR\_SECRET\_KEY

**Updates**

You should occasionally execute git pull origin master to ensure that you're using the latest version of the program.

**Usage**

The program contains two required arguments.

**Required arguments**

* **Local Directory** is the path to the local directory which contains the local files to be transferred.
* **Bucket Name** is the name of the destination S3 bucket. Please ensure that this name is unique and has not been used for other S3 buckets in your aws console.

**Optional arguments**

* **File extension** is the extensions of the files in Local Directory that need to be transfered. Extensions should be separated by ',' only. If FILES\_EXTENSION is not specified, all files in the directory are uploaded (except files whose names start with '.'). Although all file types will be uploaded to the S3 bucket, this program is meant for uploading files with csv format.

**Running the program**

Execute:

uploader **Local\_directory** **Bucket\_name** -ext **File\_extension**

Example:

python uploader.py "/Users/folder " "vital-signs" -ext "csv"

**Other information**

**Uploading coursera**

**Downloading coursera files**

You can use Coursera scheduled exports to request data exports for one or multiple courses and download them with a single command. After, downloading the required files, Data uploader to redshift cluster can be used as the next step.

**Scheduling downloads**

You can use e.g. crontab (linux, [example](https://github.com/JasperHG90/coursera-scheduled-exports-extra/blob/master/crontab.txt)) or automator (mac, [example](http://apple.stackexchange.com/questions/59532/create-automator-service-with-a-python-script)) to automate requests every week, month etc.