Steps for downloading data directly from Kaggle to Amazon S3

This tutorial demonstrates how to download data sets from Kaggle directly to Amazon S3. The goal is to stream the data directly to S3 without the data first “landing” on your laptop or on the file system of your EC2 instance. The general steps to do this are to configure the AWS Command Line Interface (CLI), install and configure the Kaggle API, then make a few edits to the Kaggle API source code to allow writing to standard output (You can read more about I/O Redirection here [[link](http://holowczak.com/introduction-linux/6/)]).

1. Set up your AWS Command Line interface CLI in your EC2 instance. Follow the instructions on slides 21, 22 and 23 of the Cloud Computing - Virtual Machines PowerPoint
2. Create a Bucket in Amazon S3 to hold your project data. For example:

$ aws s3api create-bucket --bucket project-data-XX --region us-east-2 --create-bucket-configuration \ LocationConstraint=us-east-2

where "project-data-XX" is a unique name you will give your new bucket.

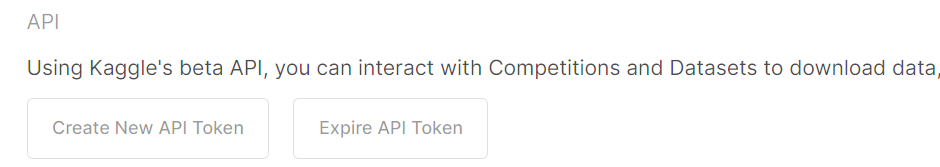
1. Install the Kaggle Command Line Interface (CLI) for Python 3. Do this as your regular user (not as root)

$ pip3 install kaggle

1. Create a new API Token in Kaggle. Go to Kaggle.com and log in. Go to the Account page.

Scroll down to the section on API

Click the button to Create New API Token.



A file kaggle.json will be downloaded.

Open up this file in Notepad or TextEdit and copy the contents to the clipboard.

1. In your EC2 instance:
   1. Make a directory for Kaggle: $ mkdir .kaggle The leading "." is important.
   2. Create a new kaggle.json file using nano editor: $ nano .kaggle/kaggle.json
   3. Paste in the text from your kaggle.json file that you downloaded from Kaggle.com. It will look something like this following but you will see your user name: {"username":"joe\_smith","key":"f0123417e63ad778000d12345"}
   4. Save the kaggle.json file and exit nano.
   5. Secure the file: $ chmod 600 .kaggle/kaggle.json
   6. Try out a kaggle cli command: $ kaggle datasets list
2. Edit the kaggle\_api\_extended.py and make two changes (shown below) to accommodate writing the file to standard output.
   1. Used the nano text editor to edit the kaggle\_api\_extended.py file:

$ nano ~/.local/lib/python3.7/site-packages/kaggle/api/kaggle\_api\_extended.py

* 1. Use the "Go To" key in Nano: ^\_ to go to line 1582 in the file.
  2. Change line 1582 from:

if not os.path.exists(outpath):

to

if not os.path.exists(outpath) and outpath != "-":

* 1. Change line 1594 From:

with open(outfile, 'wb') as out:

to

with open(outfile, 'wb') if outpath != "-" else os.fdopen(sys.stdout.fileno(), 'wb', closefd=False) as out:

* 1. Save the file and exit Nano

1. Pick a Kaggle data set you will be working with (For example I chose: totoro29/air-pollution-level. You will need the author and data set name. Use the command: kaggle datasets list to search for a data set.
2. Use the kaggle datasets download command to fetch the data set and send the file to standard output. Pipe this output to the aws s3 cp command and direct it to your S3 bucket. You also need to give it a file name. For example to download the data from the totoro29/air-pollution-level Kaggle project to an S3 bucket and file s3://project-data-XX/polution.zip use the following command:

kaggle datasets download --quiet -d totoro29/air-pollution-level -p - | aws s3 cp - s3://project-data-XX/polution.zip

1. Check your S3 bucket to see if the file was downloaded

$ aws s3 ls s3://project-data-XX/