2020 AWS EMR CHECKLIST

To help with this task, I've created a checklist. Please note that this checklist is to be used in conjunction with the POA and resources. The purpose of this checklist is to point out areas that require special attention.

Task 1

1. If you have not already downloaded python, then download version 3.6x using the graphical installer from the Anaconda site.
2. Make sure that you downloaded python version 3.7x. For OSX or Linux, at the command prompt type: python --version. For windows, type: python. For windows, if you get an error, you may need to add python to your path by following these directions (be sure that you can run python from your root directory or anywhere else from within your file system by having it in your path). If you installed python using the Anaconda distribution, and you don't have version 3.6x, then you will need to change the python version to 3.6x - the instructions are here.
3. Find a tutorial for understanding the basics of how to navigate your computer's file structure from the command line.
4. Create an AWS account. Follow the instructions in the POA that are detailed in - AWS_Account_Setup_JAN2020.pdf.
5. Install the AWS CLI (command line interface) - the instructions are here. Note: if you have python 3.6x installed (python --version), but getting AWS with python version 2.7x, then install with bundled installer for OSX, or with MSI installer for Windows.
6. Test the AWS CLI install. To test the AWS CLI install, in the command shell run: aws help. If it runs, then exit help by entering: q. If you receive a warning that "aws" is not a recognized command, you will need to follow the instructions for adding CLI to your PATHWAY. Visit the AWS CLI install pages here.
7. Configure ASW CLI. To configure AWS CLI, run the command: aws configure. See instructions here. NOTE: Be sure to enter the Region Name that you set-up (we recommended us-east-1), and set the output format to: json. You should have made a note of, or downloaded a csv file, containing the Access Key and Secret Access Key when you created the IAM User per the instructions above for creating an AWS account. If you misplaced these credentials, then you will need to recreate new credentials by returning the the IAM User set-up.
8. Install Cyberduck. Follow the instructions that are detailed in the document – How To Access S3 with CyberDuck.pdf, which can found in the Resources tab under S3 Browsing Resources.

Task 2

9. Acquire CC WET files. As a first step to getting our input addresses, visit the Common Crawl Blog http://commoncrawl.org/the-data/get-started/ and download the wet paths file for the most recent month. To do this, click on the most recent month, then click on the link beside "WET files". This will download a zipped folder with the ".gz" extension named "wet.paths.gz". Double-click on wet.paths.gz to unzip it, which will result in a text file by the same name: wet.paths. Open this text file in Sublime Text or other text editing program. Copy a single path/segment into a new tab in the text editor. Use the "replace" function to replace "crawl-data" with "s3://commoncrawl/crawl-data". Save this new tab/file as file with the ".bdf" extension. You will use the path name in an upcoming step.
10. Set up S3 buckets at AWS (script/output/debug). Note: be sure to follow AWS rules for naming these buckets, otherwise your cluster run will fail.
11. Use Cyberduck to upload the Mapper.py and Reducer.py to the script bucket.
12. Use the AWS Web Console to create an EMR cluster using the single segment that you created in a previous step. Below is an outline of the steps along with some important tips.

    a. Go to services and select EMR

    b. Create Cluster

    c. Cluster name - this will done in Advanced Options

    d. Go to Advanced Options (4 step set-up)

    e. Step 1: Software and Steps

      i. Step type: Streaming

      ii. Select: Auto-Terminate. Make sure this is selected.

      iii. Configure

        1. Select Mapper: select actual file.
        2. Select Reducer: select actual file.
        3. Input S3 Location: paste the CC WET file. (s3://commoncrawl/...).
        4. Output S3 Location: add a unique folder name (not currently in existence) to end of the path to avoid a failed cluster run. To do this, first select the folder, then click "Select". After this, add the unique folder name - do not add a "/" at the end of the unique folder name. This needs to be done for each cluster run.
        5. Arguments: No input is required - leave blank.
        6. Action on failure: "Terminate" if running a single step; "Continue" if running multiple steps.
        7. Click "Add".
        8. Click "Next" in bottom right of window - may need to expand window to see.

f. Step 2: Hardware Configuration
   i. Node type: Master, Core, Task
   ii. Instance type: accept the default instance type
   iii. Instance count: Master (1 instance); Core (2 instances - default); Task (0 instances - default). Note: When processing large numbers of segments (such as twenty or more), multiple parallel jobs should be started using the CLI and JSON job files - we'll do this when running multiple segments.
   iv. Purchasing option: On-demand (default)
   v. Click "Next".

g. Step 3: General Cluster Settings
   i. General Options:
      1. Cluster name: Name the cluster.
      2. Logging: Check and select folder.
      3. Debugging: Select.
      4. Termination protection: Select.
   ii. Tags
      1. No additional settings required.
   iii. Additional Options
      1. No additional settings required.
   iv. Click "Next".

h. Step 4: Security
   i. EC2 key pair. Leave default settings.
   ii. Select: "Cluster is visible to all IAM users in account".
   iii. Select: "Default" permissions and leave both the EC2 Security Groups and Encryption options at their default settings

i. Run Job by clicking on "Create cluster".
   i. You can access or check the status of the Hadoop Streaming job by returning to the EMR console and clicking “Refresh.” The “Status” column will first display “Starting,” then, if all steps were completed correctly, this column will display “Running” and, when the job is done it will display “Terminated, All steps complete.”
   ii. If there are errors, check the log files to determine the cause of the error. Here is a link to good AWS resource for cheking the log files.

j. Review Output
   i. Once the job has finished running, navigate to the S3 bucket (unique folder in the output bucket) and download the output to your computer through AWS S3 console, CyberDuck or via AWS CLI (see resources for download with CLI instructions).

k. Use command line to concatenate the output part files into one csv file for submission.
   i. MAC: > cat part-00000 part-00001 part-00002 part-00003 part-00004 part-00005 part-00006 > output.csv
   ii. PC: > type part-00000 part-00001 part-00002 part-00003 part-00004 part-00005 part-00006 > output.csv

l. Submit Preliminary Output

13. Use the CLI to create an EMR job. To make sure that your settings are correct using the CLI, run a cluster using only 3 WET files. Note: make sure to verify that your AWS credentials are configured correctly by typing the following at the command prompt: aws configure.
14. Create a bdf file using 3 WET files. Refer to Step 9.
15. Create a JSON file that uses the bdf file created in the previous step. Get the createJsonFilesPv3.py script that is located in the Resources section. Open in text editor and edit lines 17/18 by inputting the correct bucket names and paths. Note: you may find it helpful to put your ouput into a subdirectory for the cluster runs. To create a subdirectory, add the subdirectory name to your outputPath - e.g., "s3://output_path_name/subdirectory_name/". Save the changes you made to this file. Next, open the command shell on your computer and change the current directory to the folder that has the createJsonFilesPv3.py file and the bdf file that you created. Run the python script from the CLI: > python createJsonFilesPv3.py. Note: the json file created may have excluded the first WET file that you selected in the bdf input file.
16. Be sure to check the validity of the newly created json file using the JSONLint program. If you experience an error, you could refer the sample json file in resources for clues as to how to debug.
17. Initiate the EMR Cluster from the CLI using the script provided in the POA. The script should work for both OSX and Windows. Note: specifiying a unique folder name for the log file is optional, but could be helpful in identifying which output log is associated with a particular cluster run (be sure to follow the bucket naming rules, as discussed above). Run the script from directory that has the JSON file.
18. After initiating a successful cluster run using the CLI for the 3 WET files, repeat the process creating a bdf file that has 100 WET files, then two more bdf files that have 150-200 WET files each.
19. Consolidate the results of the EMR jobs.
    a. Create a folder and download all of the individual output *folders* from the EMR output to this folder. Note: Keep the folders for each EMR job step, you don't have to pull the Part files out of them.
    b. Download the concatenatepv3.py file and put it in the same folder where you saved all your EMR output folders and run this script from the command prompt. The script will ask you for the root location from which to start the walk of directories - leave blank (it defaults to the current directory). If the script runs successfully, you will have two new files in your folder: 'concatenated_websites.csv' and 'concatenated_factors.csv'.
    c. Open concatenated_factors.csv and inspect the number of instances - you need at least 20,000 instances. If you are short of 20,000 instances, you may need to search additional WET files using the steps above. You can run up to 255 steps/Wet files at a time. If you have at least 20,000 instances, move on to the next step.
    d. Rename 'concatenated_factors.csv' to LargeMatrix.csv. Zip and submit your LargeMatrix file.