Decode HR

Server Deployment User Guide

Contents

[1 Introduction 4](#_Toc40090999)

[2 Section Summary 5](#_Toc40091000)

[3 Deployment Process 6](#_Toc40091001)

[4 Folders and files Creation 11](#_Toc40091002)

[4.1 GoogleNews Folder 12](#_Toc40091003)

[4.2 Twitter Folder 13](#_Toc40091004)

[4.3 GoogleNews python files 14](#_Toc40091005)

[5 Linux Cronjob Configuration 15](#_Toc40091006)

[5.1.1 Scheduling timing Configuration in python files 15](#_Toc40091007)

[6 Program Execution Procedures (Linux) 18](#_Toc40091008)

[6.1 Main Program to run: (All these programs will be run in the Server\_schedule\_process.py) 18](#_Toc40091009)

[6.2 For GoogleRSS HR news scrapping procedures: 18](#_Toc40091010)

[6.3 For Creating GoogleRSS HR news dataset procedures: 23](#_Toc40091011)

Table of Figures

[Figure 3‑1 : Software – Setup Python App 6](#_Toc40445864)

[Figure 3‑2 : Python – Create application 6](#_Toc40445865)

[Figure 3‑3 : Create Application Form 6](#_Toc40445866)

[Figure 3‑4 : Successful Result of Application Creation 7](#_Toc40445867)

[Figure 3‑5 : Advance – Cronjob 7](#_Toc40445868)

[Figure 3‑6 : Cronjob Weekly task configuration option 8](#_Toc40445869)

[Figure 3‑7 : Scheduling Configuration details 8](#_Toc40445870)

[Figure 3‑8 : Cronjob listing 9](#_Toc40445871)

[Figure 3‑9 : Command copy for activation of Python Virtual env 9](#_Toc40445872)

[Figure 3‑10 : Advance - Terminal 9](#_Toc40445873)

[Figure 4‑1 : Main Dashboard Folders 11](#_Toc40445874)

[Figure 4‑2 : GoogleNews subfolders 13](#_Toc40445875)

[Figure 4‑3 : Twitters subfolders 14](#_Toc40445876)

[Figure 5‑1 : Linux Cronjobs 15](#_Toc40445877)

[Figure 5‑2 : Scheduling Python Code - RunTaskPeriodically 16](#_Toc40445878)

[Figure 5‑3 : Advance – Terminal 17](#_Toc40445879)

[Figure 5‑4 : Killing previous cronjob process 17](#_Toc40445880)

[Figure 6‑1 : Google\_search\_Category.txt input file 18](#_Toc40445881)

[Figure 6‑2 : Fie Manager in Server 19](#_Toc40445882)

[Figure 6‑3 : Navigate to Input folder 19](#_Toc40445883)

[Figure 6‑4 : Uploading the selected text file to Server 20](#_Toc40445884)

[Figure 6‑5 : Data scrapped from GoogleRSS news site in JSON Format 21](#_Toc40445885)

[Figure 6‑6 : GoogleRSS news site scrapping output console 21](#_Toc40445886)

[Figure 6‑7 : GoogleRSS News site scrapping log files 22](#_Toc40445887)

[Figure 6‑8 : Stopwords text file 23](#_Toc40445888)

[Figure 6‑9 : Diagram folder 24](#_Toc40445889)

[Figure 6‑10 : Output console of Create\_Google\_news\_dataset\_by\_Category.bat 24](#_Toc40445890)

[Figure 6‑11 : Create\_Google\_news\_dataset\_by\_Category datasets folder 25](#_Toc40445891)

[Figure 6‑12 : GoogleRSS News site scrapping log files 25](#_Toc40445892)

# Introduction

This is the detail procedures description on how to deploy and configure the python environment in the Server.

# Section Summary

This section is a summary description of each section.

|  |  |
| --- | --- |
| Section | description |
| deployment process | Description of the Server Deployment Procedures |
| Folders and files Information | Details of the files and folder structures |
| Linux Cronjob Configuration | Details of how to configure the cronjob scheduling |
| Program Execution Procedures (Linux) | Details of the program execution steps for Linux Platform |
|  |  |
|  |  |

# Deployment Process

source /home/dchrahos/virtualenv/DCHR\_DashBoard/3.7/bin/activate && cd /home/dchrahos/DCHR\_DashBoard

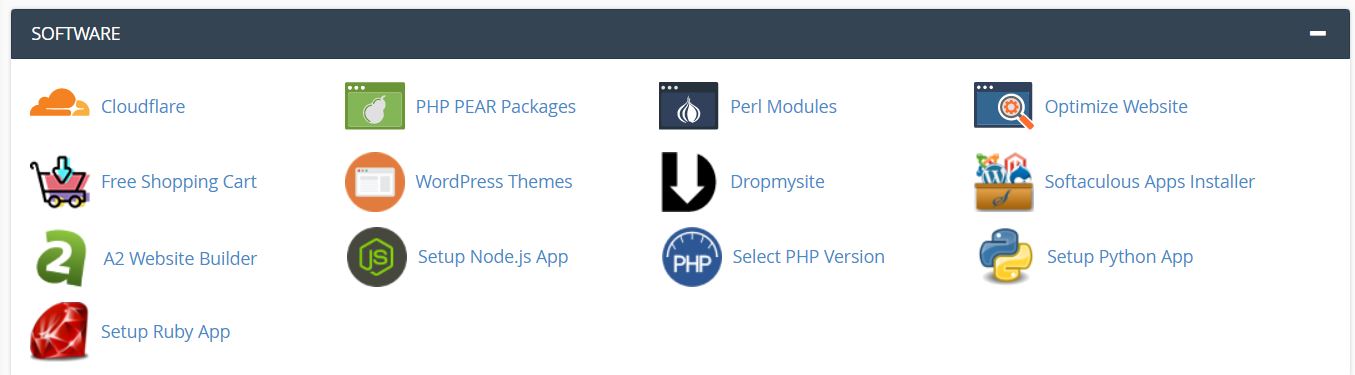


Figure ‑ : Software – Setup Python App

1. Login to the link : <http://www.dchr.a2hosted.com/cpanel> username : dchrahos / *D9vSD3bFw8()5m*
2. Go to Software section and click on the “Setup Python App

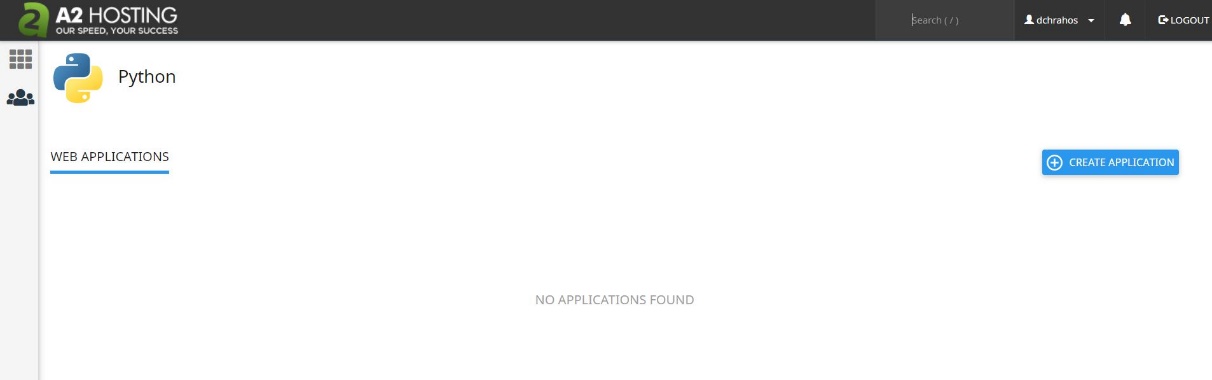


Figure ‑ : Python – Create application

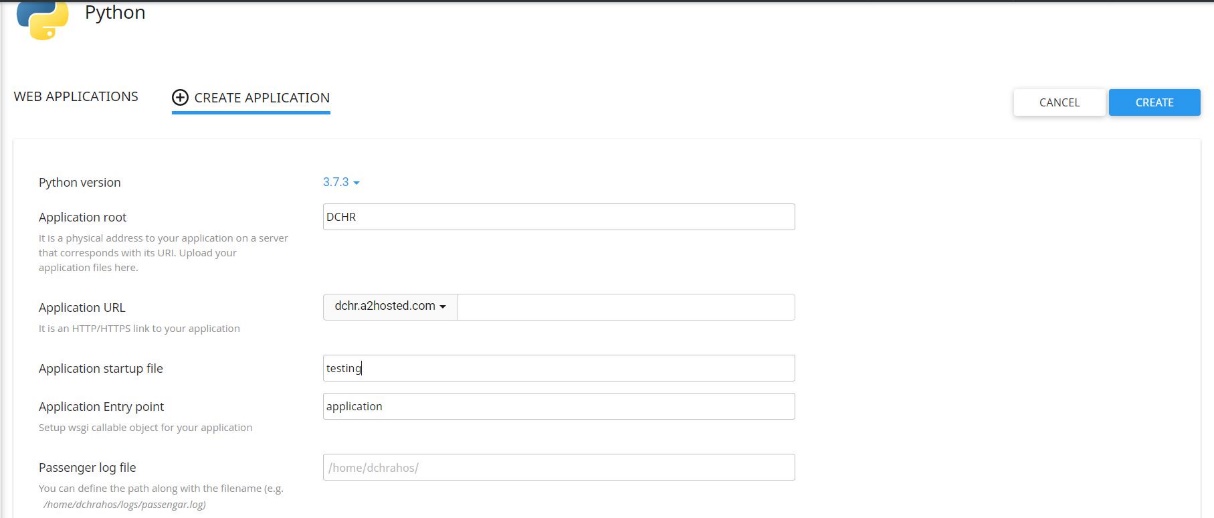
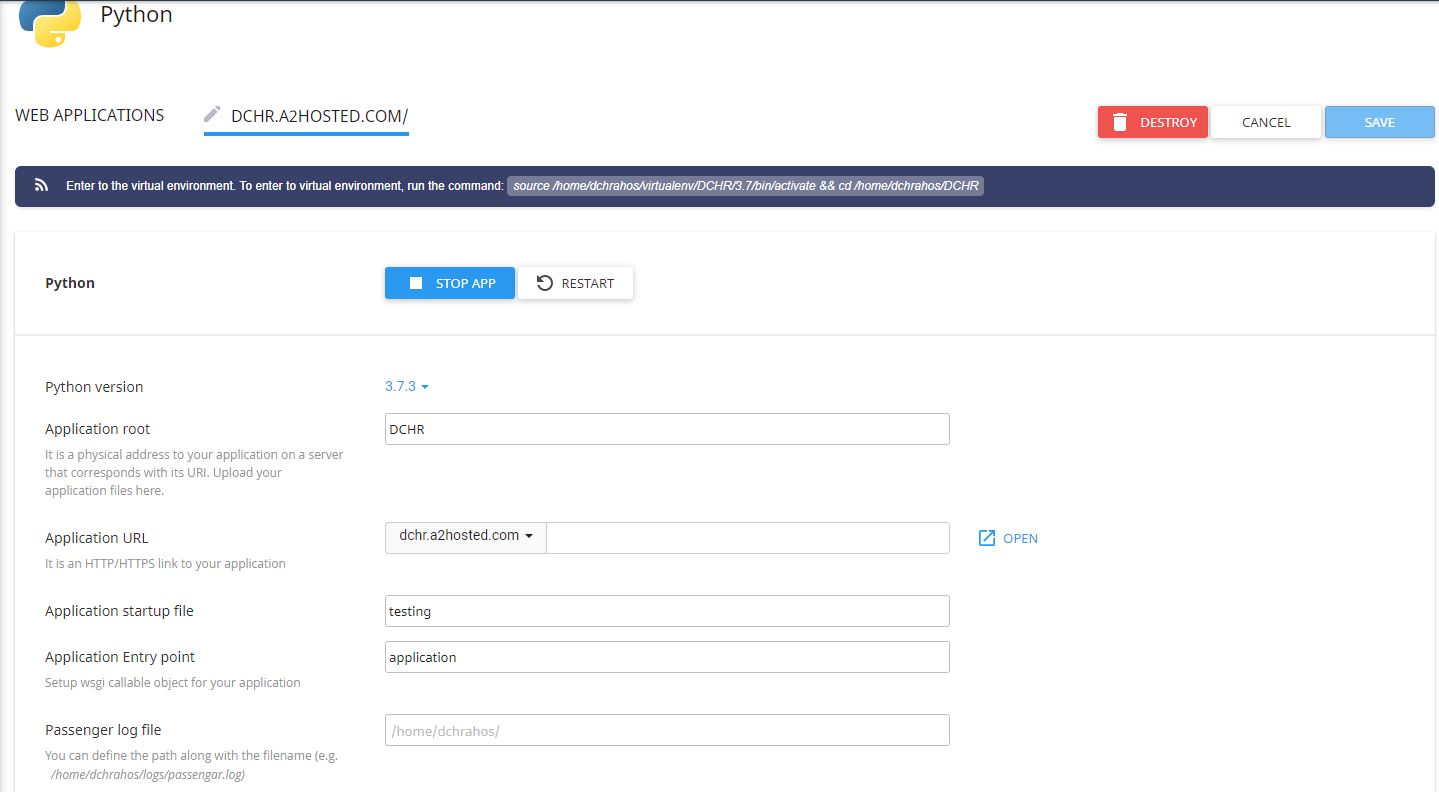
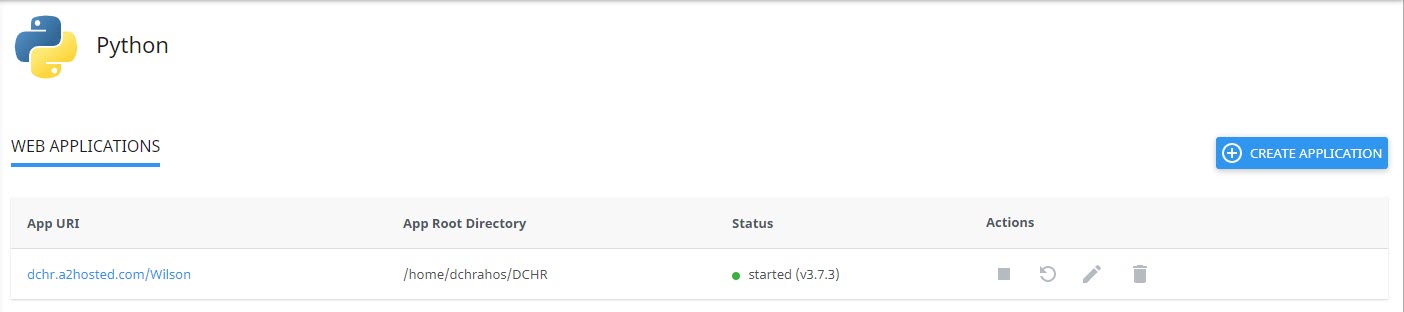


Figure ‑ : Create Application Form

1. Create Python Application:
   1. Choose the latest python version, in this case version 3.7.3
   2. Application Root – The folder name to store all the Virtual environment of python installations
   3. Application URLs – Domain name for the server
   4. Application Startup file – A test python file name for testing if the creation of the application is successful
   5. Application Entry point – he Function name in the python startup file
   6. Click on the link created eg: dchr.a2hosted.com/DCHR\_Dashbaord to test





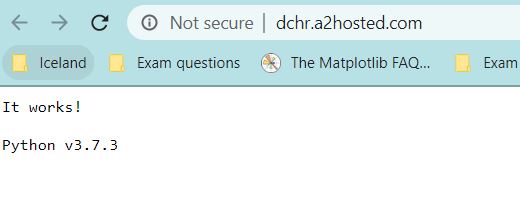


Figure ‑ : Successful Result of Application Creation

1. Cronjob
   1. Go to Advance section and click on “Cronjobs”

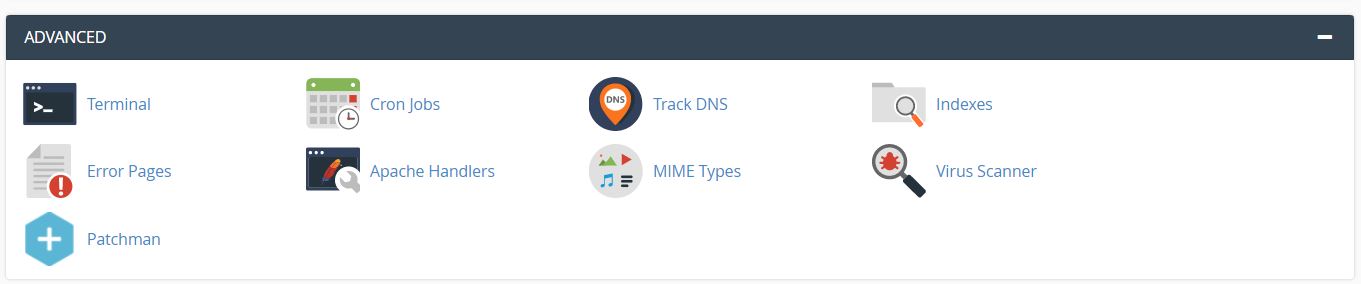


Figure ‑ : Advance – Cronjob

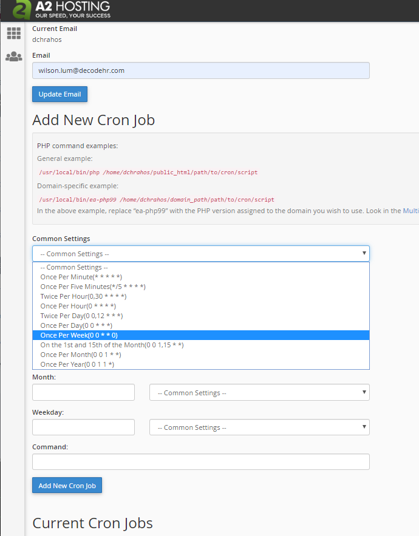


Figure ‑ : Cronjob Weekly task configuration option

* 1. With command :   
     source /home/dchrahos/virtualenv/DCHR\_Dashboard/3.7/bin/activate && cd /home/dchrahos/dashboard && python Server\_schedule\_process.py

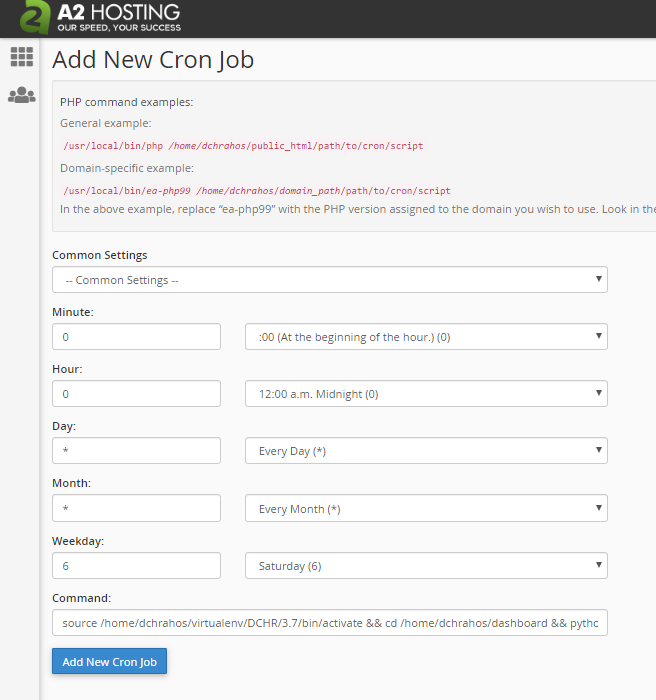


Figure ‑ : Scheduling Configuration details

* 1. Cronjob listing:



Figure ‑ : Cronjob listing

* 1. Take note if the cronjob is edited and reschedule, the previous python process will need to be killed if not the process will take up resources. Use “ps -ef | grep python” to check what is the process id for the previous process and then use “Kill -9 <pid>” to kill the old process

1. Copy the command to use in the terminal command line to activate the python env as show in the next step.

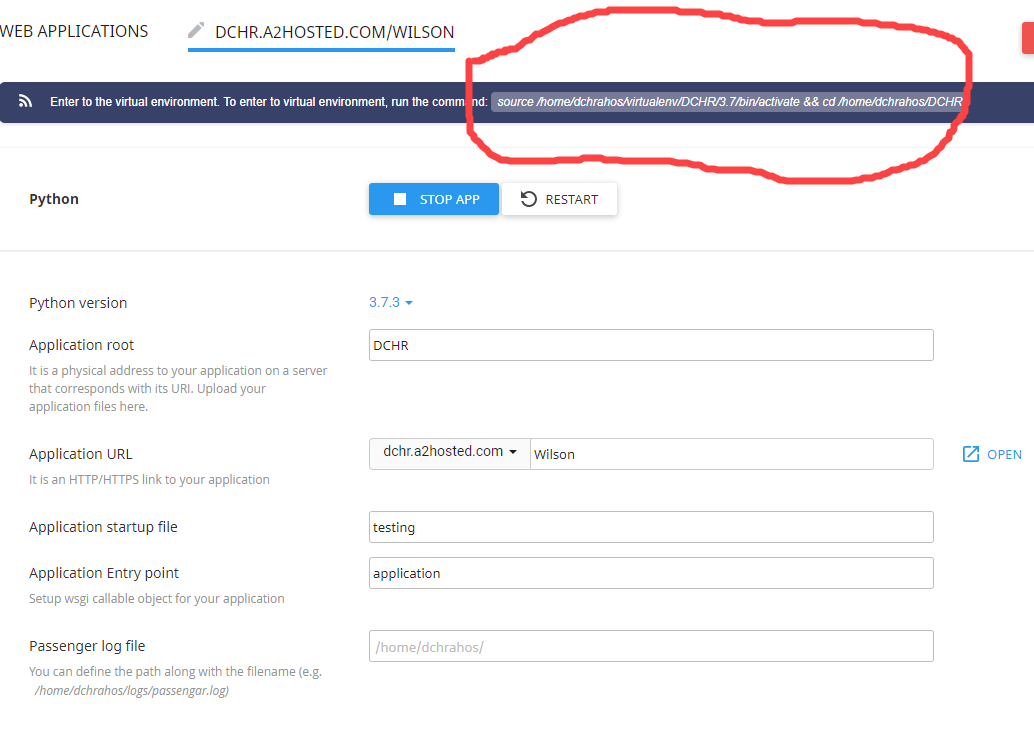


Figure ‑ : Command copy for activation of Python Virtual env

1. Activation of Virtual Environment
   1. Go to Advance and click on the “Terminal”

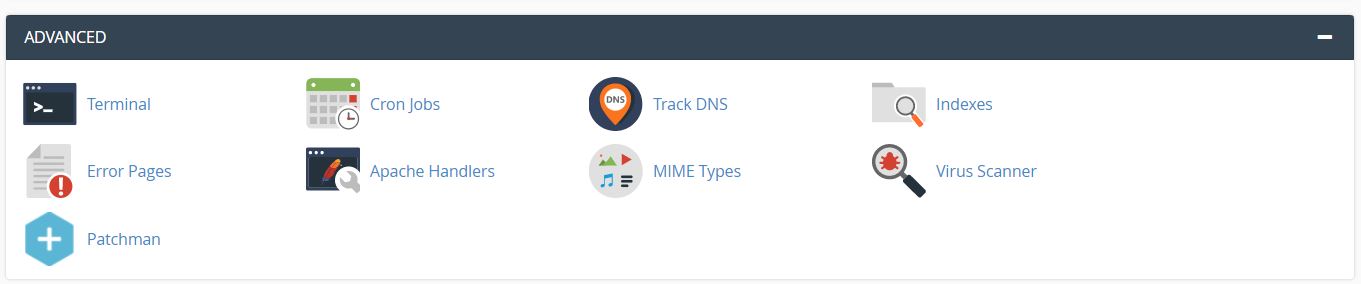
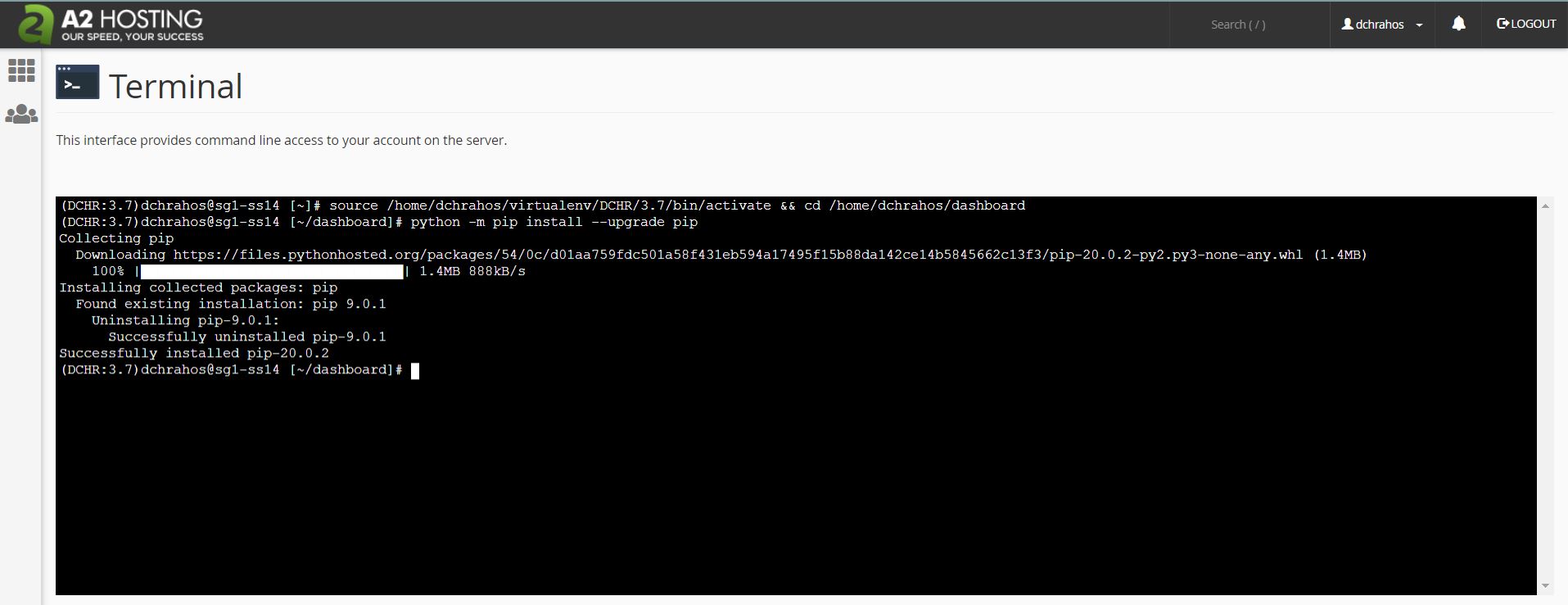
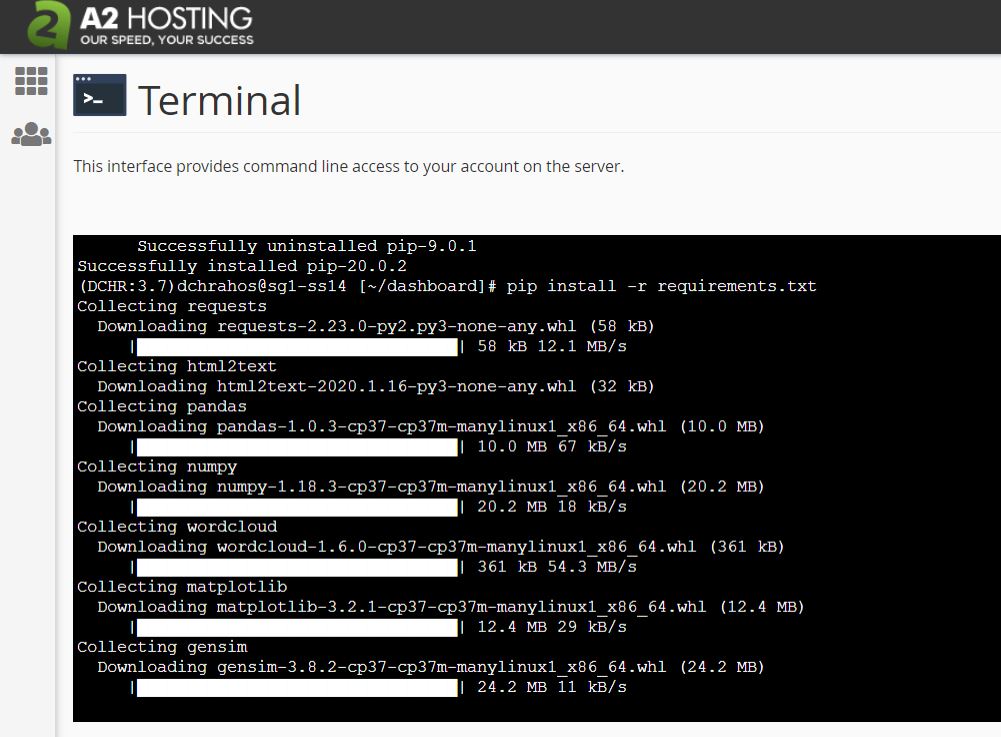


Figure ‑ : Advance - Terminal

* 1. Follow the “pip installation.txt” instruction.
  2. First is “python -m pip install --upgrade pip” to make sure we have the latest pip



* 1. Next is “pip install -r requirements.txt” to install the required python library or packages



* 1. Next will be “python -m spacy download en\_core\_web\_sm” . This will take sometime to install as it requires to download a huge data file
  2. Next will be “python install\_nltk\_spacy\_package.py” to install and download the package data

# Folders and files Creation

The main folders information as follow: (As shown in Figure 5‑1). It is required to create the following folders structure.

1. GoogleNews Folder
   1. GoogleRSS news scrapping, pre-process and Power BI files
2. Twitter Folder
   1. Twitters news scrapping, pre-process and Power BI files
3. Readme.txt
   1. Summary of the installations and the procedures of how to run the programs
4. Schedule\_process.log
   1. This is the log file to keep logs of the running the scheduling of each python codes
5. Server\_schedule\_process.py
   1. This is to run the schedule python code for Linux Server platform in Support Board
6. Log folder
   1. For storing the schedule log process

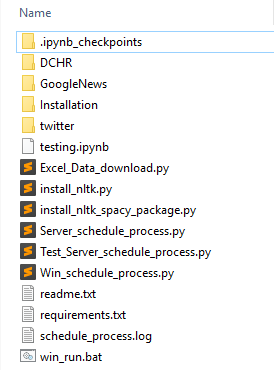


Figure ‑ : Main Dashboard Folders

## GoogleNews Folder

The folders for GoogleNews is as follow: (As shown in Figure 4‑2)

1. Doc
   1. This folder stores the user guide word doc
2. Log
   1. This folder stores all the log text files created when running all the python codes
   2. Log files are named with the respective dates
3. Data
   1. This folder stores the excel data files created by the web scrapping and pre-processed python codes
   2. These excel files are used and input to Microsoft Power BI for the dashboard display
   3. The same excel file is duplicated with the second file name added with dates. Th original file is used by Power BI while the second file is for backup and logging purpose. Eg “DCHR\_Google\_Product.xlsx” is used by Power BI while “DCHR\_Google\_Product\_Apr-08-2020.xlsx” is used for logging and backup purpose.
4. Diagram
   1. This folder stores the bigram (2 words) and trigram (3 words) word cloud picture created from the python codes
   2. These pictures can be used for web publishing
   3. These pictures are used as an input to create format and save into excel file sheet name “Images” in the dataset folder name “dashboard/data”. They are display in the Power BI “2 word wordcloud” and “3 word wordcloud” tab
5. Input – The following underlined brown text files have to be uploaded to this folder for python codes to read
   1. This folder stores the text file where user can edit and input their desire keywords for their search engines as follow:
      1. Google\_search\_Category.txt (For the HR category keywords)
      2. Google\_search\_Keyword.txt (For any HR trend keywords)
      3. Google\_search\_Product.txt (For HR software Product name keywords)
   2. Stopwords can be edited and input to the following:
      1. stopwords.txt
   3. The number range of topics (which is used to test what is the optimised number of topics to used)
      1. number\_of\_topics.txt
      2. First number is the min number of topics to processed
      3. Second number is the maximum number to processed
      4. If this file is not found the default min of 3 and max of 5 will be used
      5. This code will take about min of 1-2 hours (Under PC Windows with 16G RAM and 6 Core Intel 4.28GHz CPU) to run depending on the size of the data been processed and the range of number of the topics.
6. Topic
   1. This is the output HTML file for number of topics visualisation
   2. File naming convention: Eg “GoogleNews\_topic\_no\_4\_Apr-08-2020.html” refer to number of topics is 4 processed on date Apr-08-2020.

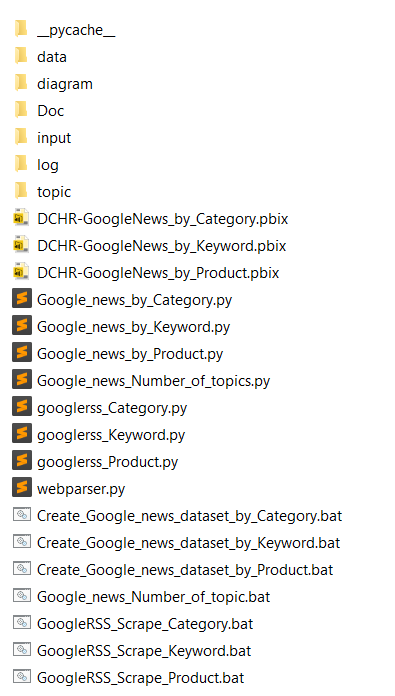


Figure ‑ : GoogleNews subfolders

## Twitter Folder

The folders for twitter is as follow: (As shown in Figure 4‑3)

1. Doc
   1. This folder stores the user guide word doc
2. Log
   1. This folder stores all the log text files created when running all the python codes
   2. Log files are named with the respective dates
3. Data
   1. This folder stores the excel data files created by the web scrapping and pre-processed python codes
   2. These excel files are used and input to Microsoft Power BI for the dashboard display
   3. The same excel file is duplicated with the second file name added with dates. The original file is used by Power BI while the second file is for backup and logging purpose. Eg “DCHR\_Influencer\_tweets.xlsx” is used by Power BI while “DCHR\_Influencer\_tweets\_Apr-08-2020.xlsx” is used for logging and backup purpose.
4. Input – The following underlined brown text files have to be uploaded to this folder for python codes to read
   1. This folder stores the text file where user can edit and input their desire keywords for their search engines as follow:
      1. Twitter By Topic List.txt (For the Twitter search by Topic keywords)
      2. Twitter Infuencer List.txt (For the Twitter search by Influencer screen name)
   2. Twitter access and authentication tokens can be edited and input to the following:
      1. Twitter\_Token.txt

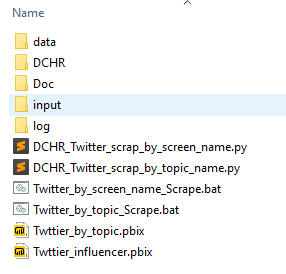


Figure ‑ : Twitters subfolders

## GoogleNews python files

The files for GoogleNews which are required to upload to GoogleNews Folder is as follow:

1. For web scrapping:
   1. webparser.py
   2. googlerss\_Category.py
   3. googlerss\_Keyword.py
   4. googlerss\_Product.py
2. For Dataset creation:
   1. Google\_news\_by\_Category.py
   2. Google\_news\_by\_Keyword.py
   3. Google\_news\_by\_Product.py
3. For number of topics optimisation: (This should run in the low peak hours overnight as it will take min of 1-2 hours (for 3-5 topics) depending on data size the range of number topics to test)
   1. Google\_news\_Number\_of\_topics.py

# Linux Cronjob Configuration

## In Linux Server platform scheduling, cronjob is used.

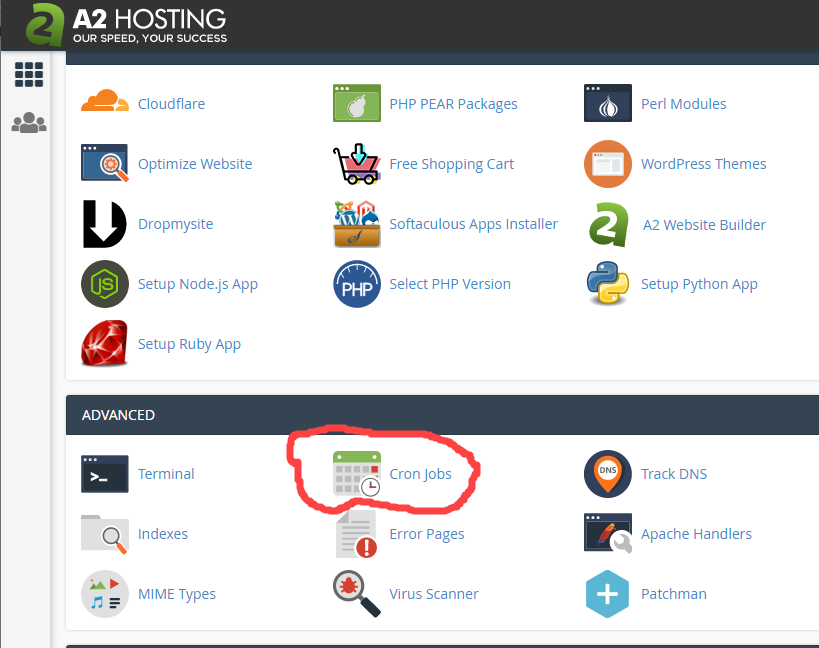


Figure ‑ : Linux Cronjobs

## Scheduling Configuration details:

### Scheduling timing Configuration in python files

Generally the timing configuration is similar for both Windows and Linux Platform. For more details of how to amend the timing and day of the scheduling, you can read at the following links:

<https://www.geeksforgeeks.org/python-schedule-library/>

You edit the python code file “Win\_schedule\_process.py” (Windows) & “Server\_process\_schedule.py (Linux) and go to the function name RunTaskPeriodically().

Eg. For “schedule.every().monday.at('14:25').do(twitter\_by\_topic\_scrap)” as the first starting program, you can change to :

schedule.every().day.at('10:25').do(twitter\_by\_topic\_scrap)

Make sure between each program, this is some time allowance given for the previous program to run. Recommended will be at least 1.5 hours between each program. For Twitter under PC windows with 16G RAM and 6 Core 4.28GHz CPU will only take about 10-15 mins to run. Sometimes it may take extra of 15 mins due to twitter max scrapping and hence require to sleep for 15 mins.

Under server linux platform, it may take longer time depends on the subscription of the Hardware resources. Hence about 1.5 to 2 hours in time difference between each program is recommended.

**Note : Once the schedule timing has changed, make sure the cronjob schedule has to be align and updated and rerun.**

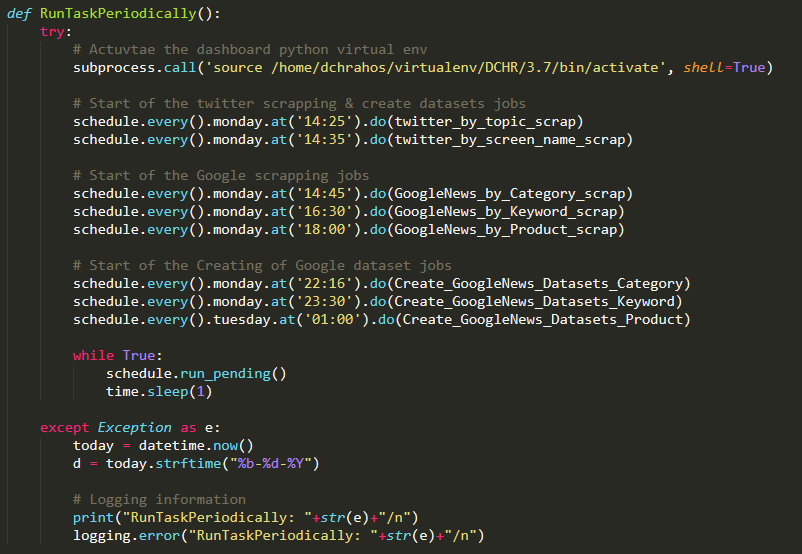


Figure ‑ : Scheduling Python Code - RunTaskPeriodically

## Editing cronjob and schedule

Editing and changing the schedule dates and time require both the cronjob and the Server\_schedule\_process.py to be align in the timing planning. After changes is done, the previous cronjob process is required to be manually killed by the following figures shown.

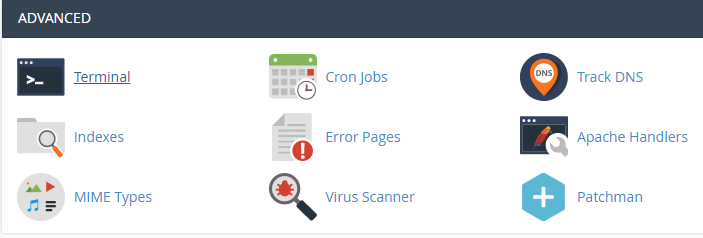


Figure ‑ : Advance – Terminal

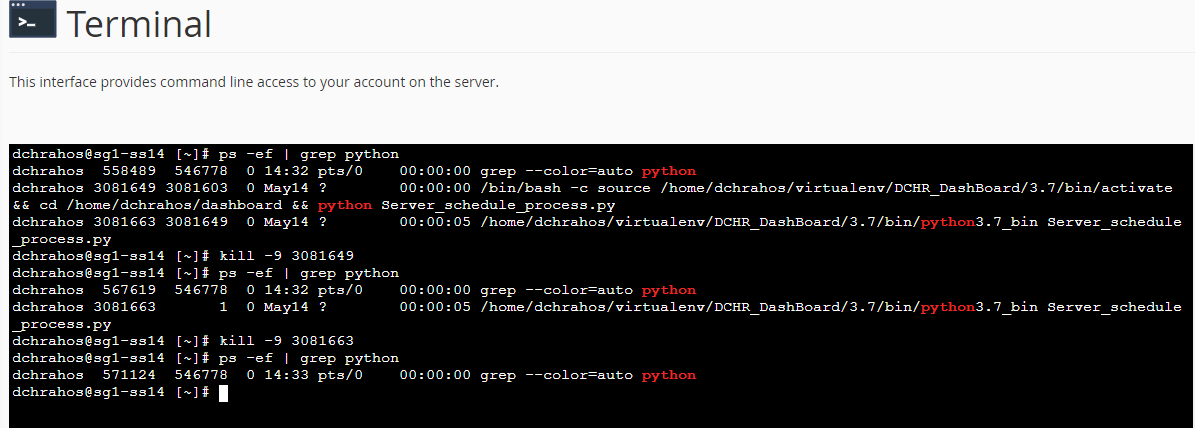


Figure ‑ : Killing previous cronjob process

# Program Execution Procedures (Linux)

In this section, it describes in details on the procedures of the python code programs being run. However, this steps 1 & 2 are **already automated in the Server\_schedule\_process.py** which will run as per the cronjob.

## Main Program to run: (All these programs will be run in the Server\_schedule\_process.py)

1. Run function - **GoogleNews\_by\_Category\_scrap()**
   1. This is to download all googlerss news related to the keywords selected in the “input\Google\_search\_Keyword.txt”.
   2. Need to input any keywords to this text file for desirable keywords to scrape
2. Run function - **Create\_GoogleNews\_Datasets\_Category()**
   1. This is to create dataset used by the Microsoft Power BI tools for Data Visualisation purpose
   2. Optional to input to the stopwords into the “input\stopwords.txt” file for undesirable words to be display in the wordcloud diagram
3. Run the “**DCHR-GoogleNews\_by\_Category.pbix**” to open out this Microsoft Power BI dashboard file.
   1. This will display the wordcloud and related data from the excel dataset created by step 2.
4. The same procedures applies for the both Keywords and Products

## For GoogleRSS HR news scrapping procedures:

For all the procedures as explained in the following will also applies to both the Keywords and Products. Eg if running function for Category keywords, then the procedure uses eg GoogleNews\_by\_Category\_scrap(). If the Product keywords is the data to scrape, then GoogleNews\_by\_Product\_scrap() should be used instead.

1. Run function - GoogleNews\_by\_Category\_scrap().
   1. This scrapper will scrape news from Google RSS site
   2. It will read the keywords to scrape from input folder “input\Google\_search\_Category.txt”
   3. For new keywords to scrap, need to input into this text file line by line

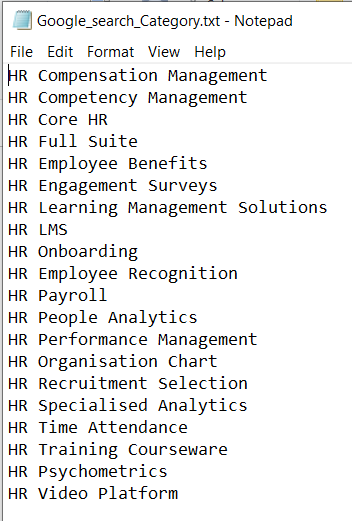


Figure ‑ : Google\_search\_Category.txt input file

1. Uploading the text file to Server
   1. Select the File Manager

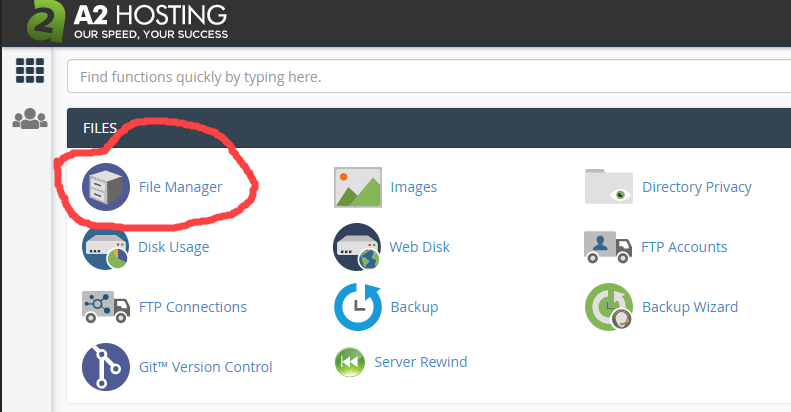


Figure ‑ : Fie Manager in Server

* 1. Navigate to the GoogleNews/input folder and select the Upload from the top Menu as shown:

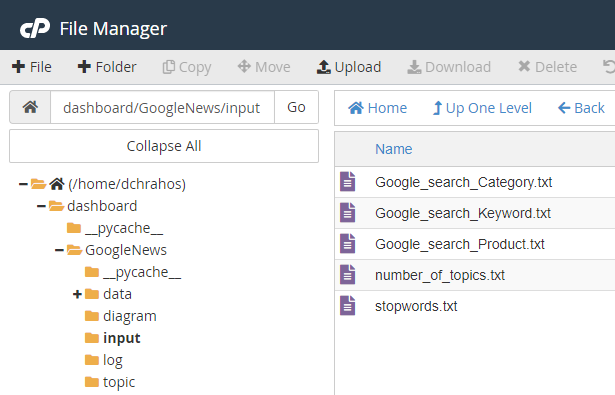
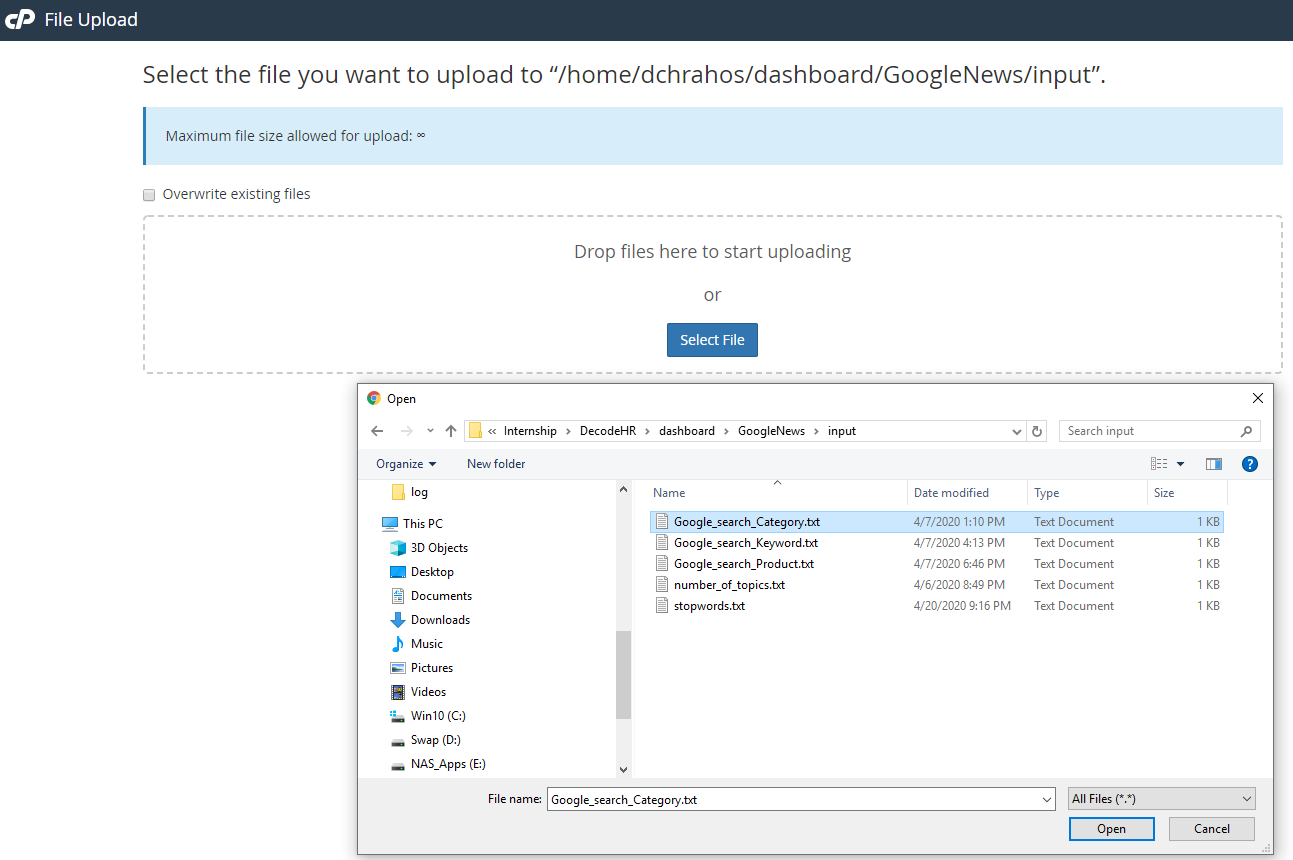


Figure ‑ : Navigate to Input folder

* 1. Select the respective text file to upload:



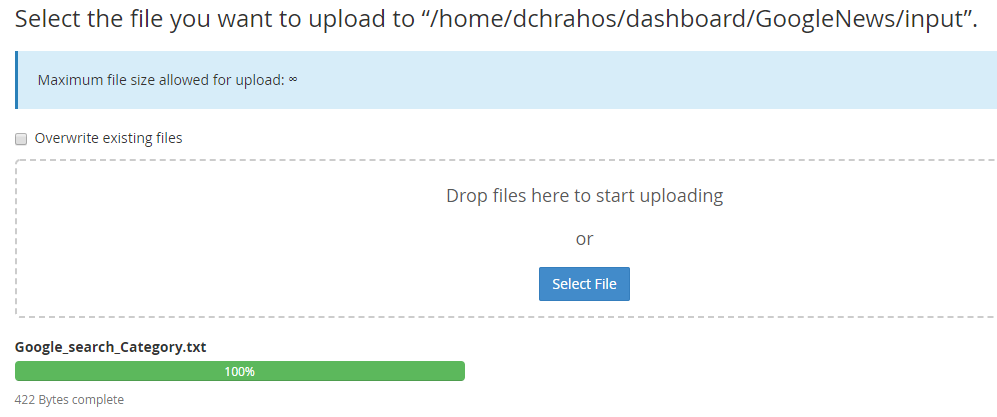


Figure ‑ : Uploading the selected text file to Server

1. After each keyword is scrapped, the data is save into json format with the keyword names. Eg scrapping keyword HR Core HR, the data will be saved into data\Category\HR Core HR.json.

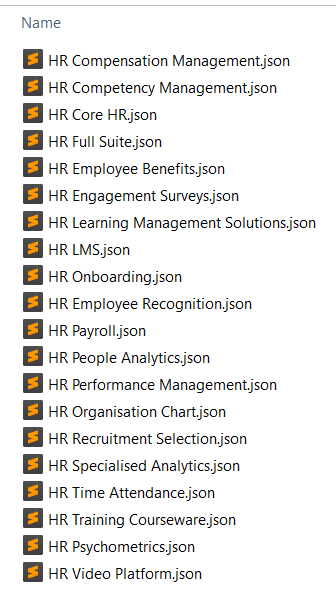


Figure ‑ : Data scrapped from GoogleRSS news site in JSON Format

1. Output console: (If the program is run from cronjob, this console will not be available)

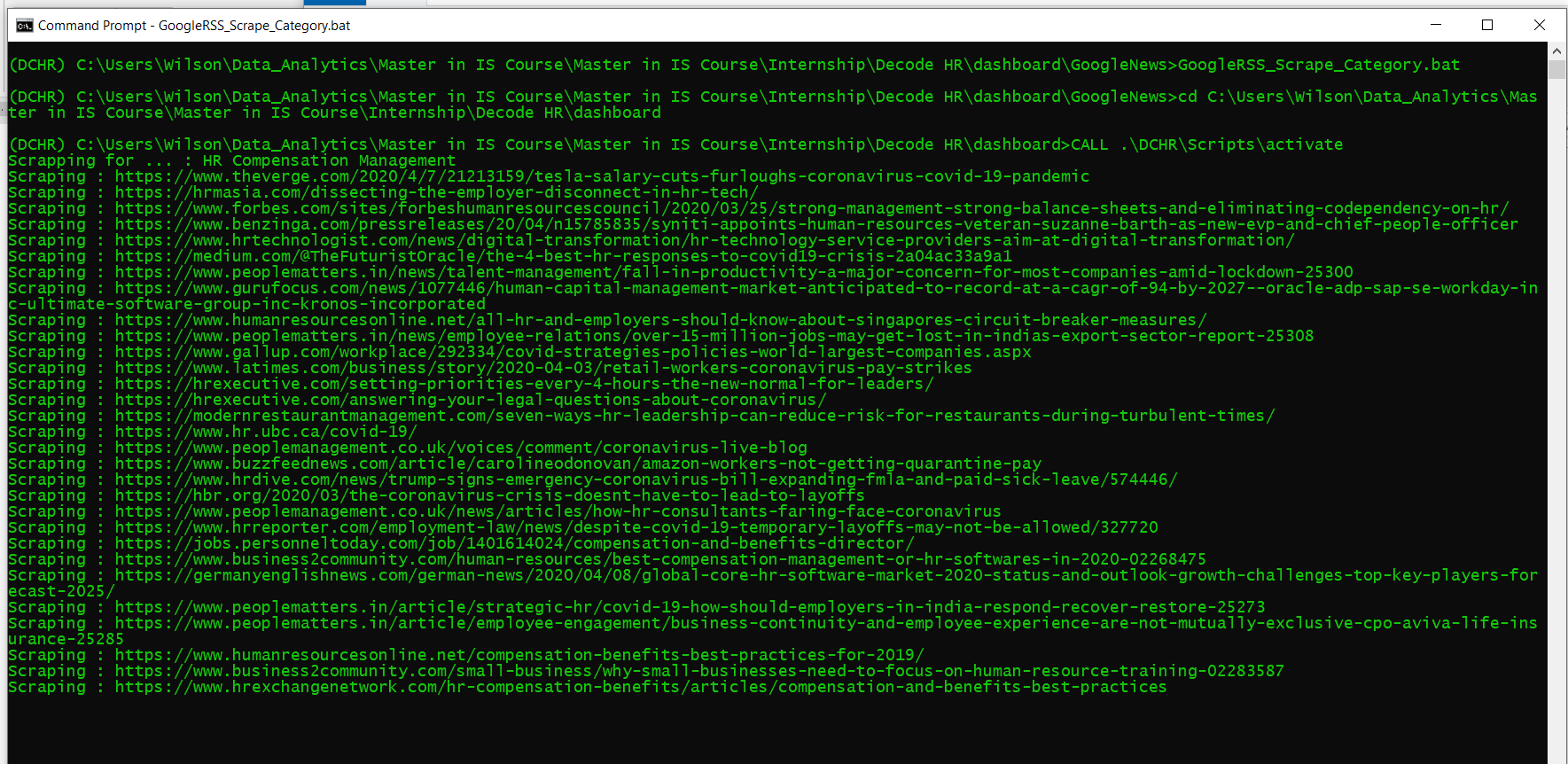


Figure ‑ : GoogleRSS news site scrapping output console

The output console of the GoogleRSS News site scrapping will show the sites it is scrapping from.

1. Log files will be created in the log folder “log\ DCHR\_Scrape\_Google\_news\_by\_Category\_Apr-13-2020.log”:

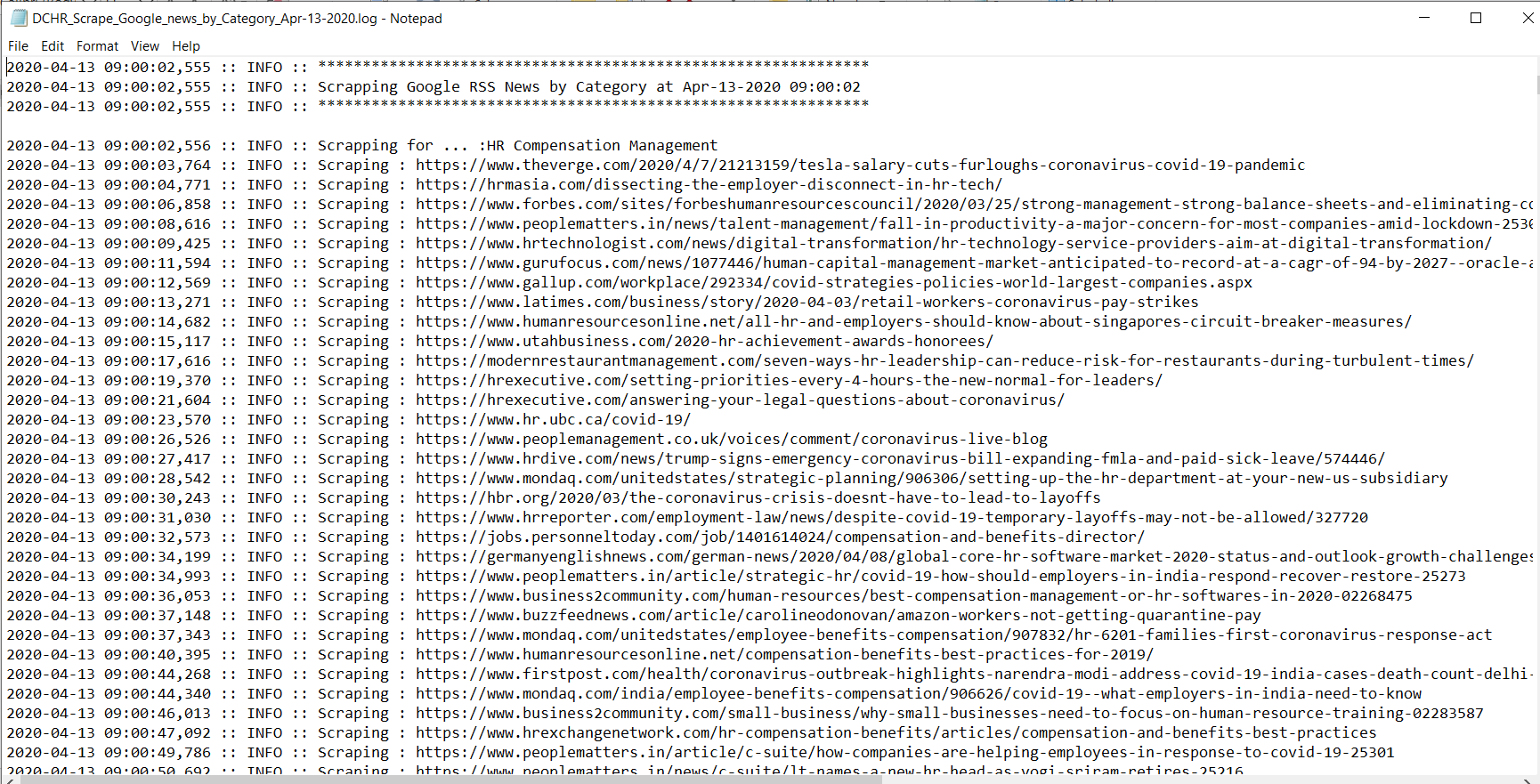


Figure ‑ : GoogleRSS News site scrapping log files

## For Creating GoogleRSS HR news dataset procedures:

1. Run batch file - Create\_Google\_news\_dataset\_by\_Category.bat
   1. This will create dataset in excel format from all the json files created from the GoogleRSS\_Scrape\_Category.bat run in data folder as shown in Figure 6‑2.
   2. It will read the stopwords text file from to scrape from input folder “input\stopwords.txt”. This contains the words that are undesirable to show up in the wordcloud.
   3. For new stopwords, need to input into this text file line by line

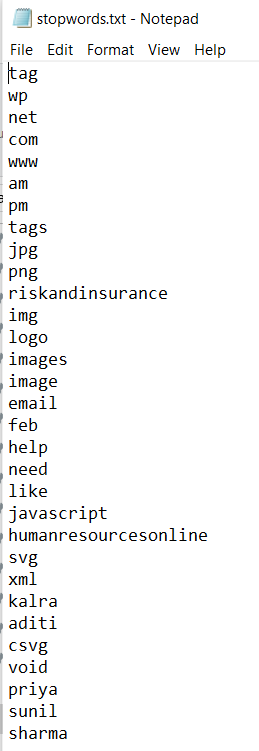


Figure ‑ : Stopwords text file

1. For every HR category data, bigram (2 words wordcloud) and trigram (3 words wordcloud) diagram will be generated in jpg format store in the diagram folder. Eg diagram\ WordCloud\_DCHR\_Bigram\_HRTECH.jpg.

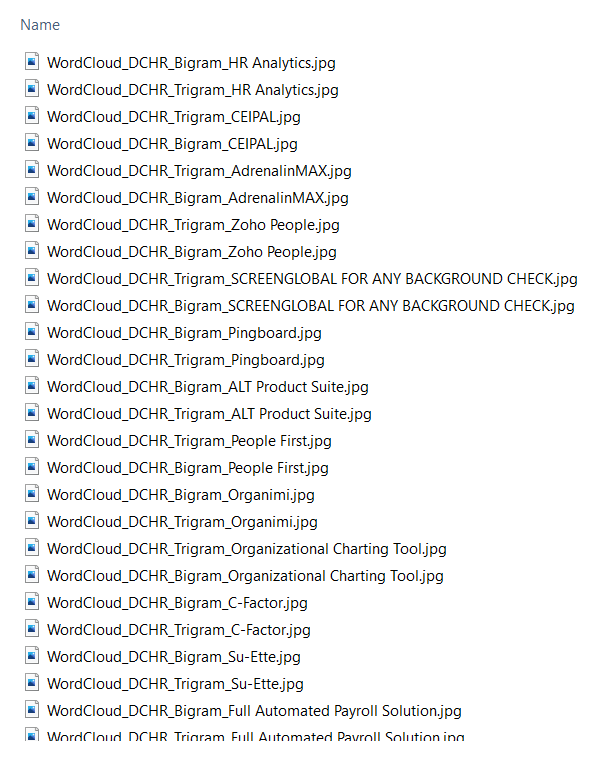


Figure ‑ : Diagram folder

1. All these jpg will not be used as input to the Power BI due to slow processing for displaying and difficult for selection reference. These jpg files will be converted to base64 format and save into excel file as another database for Power BI to point to. These data will be saved into folder data/DCHR\_Google\_Category.xlsx in Image worksheet.
2. Output console will also shows the progress of the program execution: (If the program is run from cronjob, this console will not be available)

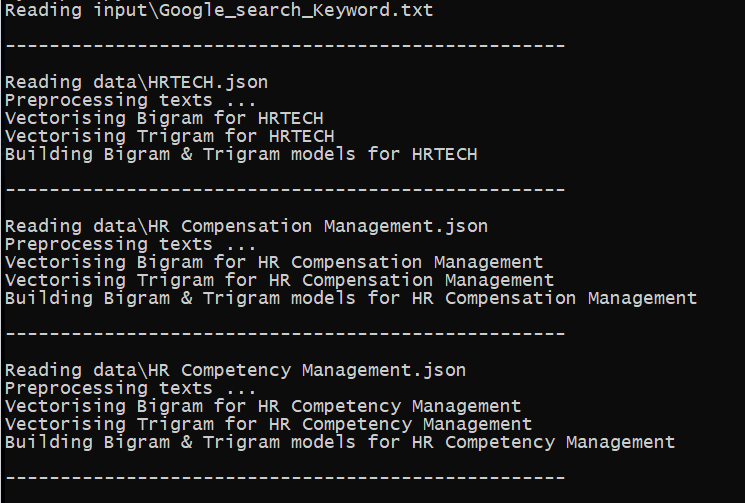
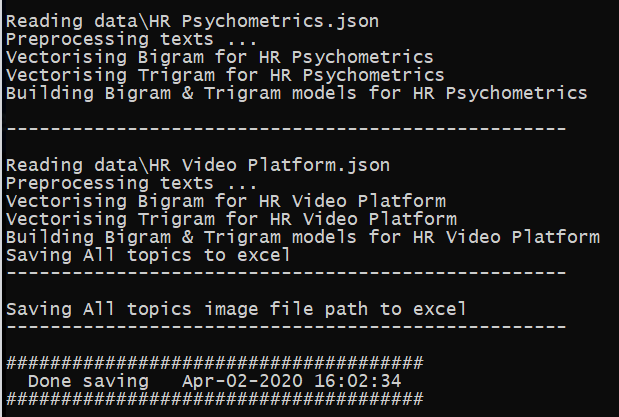
 

Figure ‑ : Output console of Create\_Google\_news\_dataset\_by\_Category.bat

1. Each of the HR category json files data will be processed and converted to a excel file saved into data/ DCHR\_Google\_Category.xlsx and DCHR\_Google\_Category\_Apr-08-2020.xlsx.

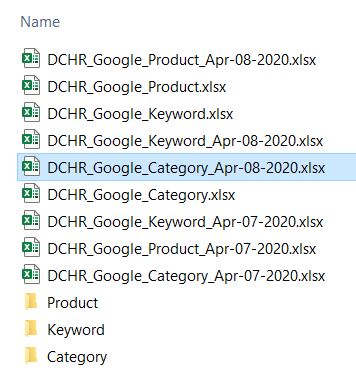


Figure ‑ : Create\_Google\_news\_dataset\_by\_Category datasets folder

1. Log files will be created in the log folder “log\ DCHR\_Scrape\_Google\_news\_by\_Category\_Apr-08-2020.log”:

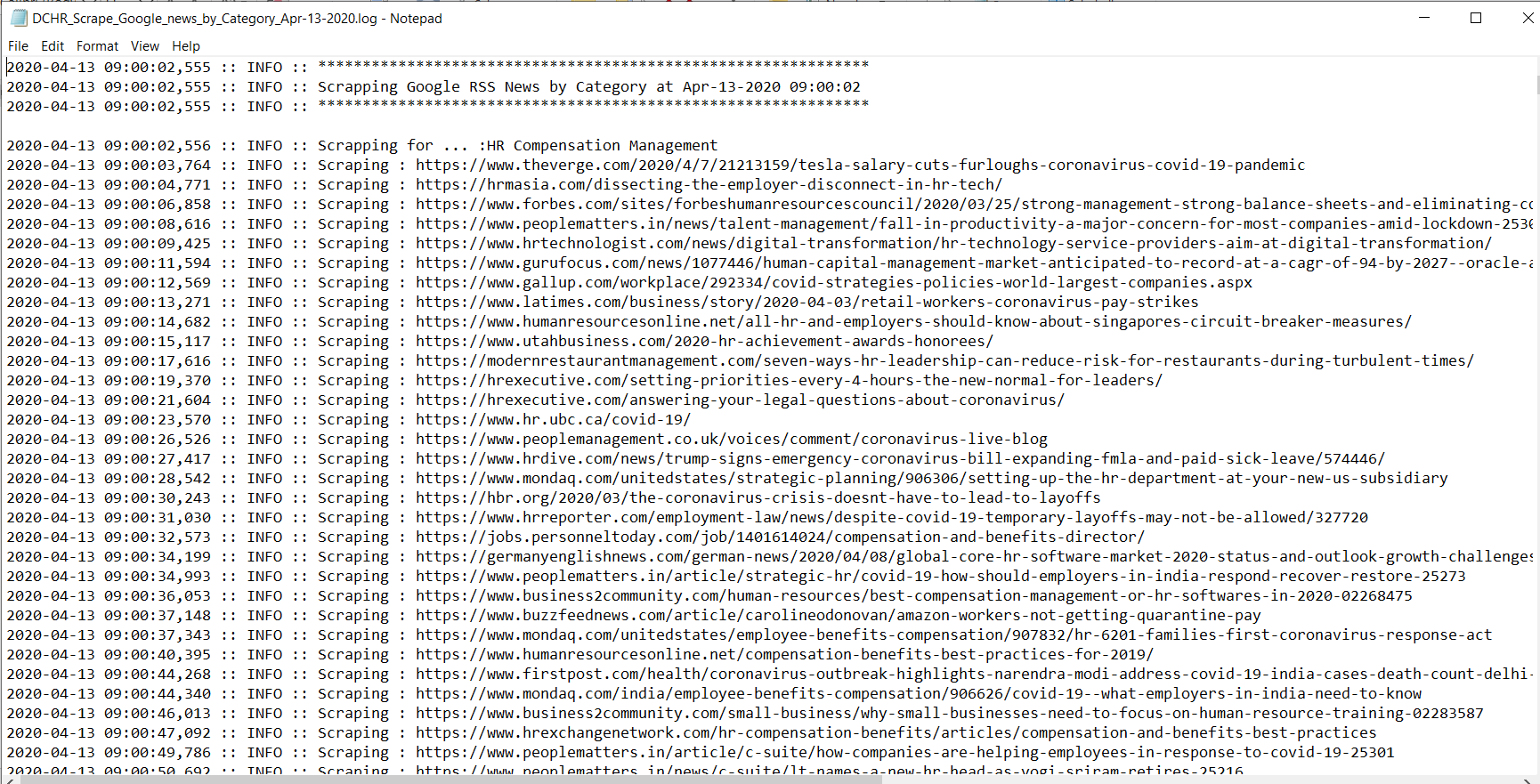


Figure ‑ : GoogleRSS News site scrapping log files