AWS with Python, Jupyter Notebooks and MRJob

Cosmin Stamate <cosmin@dcs.bbk.ac.uk>

Overview

- Create a Microsoft Azure Notebooks account
  - https://notebooks.azure.com/
- Or, use a local Python distribution
  - Anaconda (if you are using your own machine)
    https://www.anaconda.com/
  - WinPython (if you are in the lab, as it is portable and does not require admin rights)
    http://winpython.github.io/ [3.6 Qt5 64bit]

- Install MRJob Python library
  - https://github.com/Yelp/mrjob
- Create and test wordcount locally and deploy script to AWS

Microsoft Azure Notebooks account

- Go to: **Sign In**
- Login with your Microsoft account (or create one if you don’t have one)
- Go to **Libraries**
- Click on: **+ New Library**
- In Library name put: **Cloud Computing**
- Library ID: **cloudcomputing**
- Press **CREATE**

Great, you now have your library where you can create Jupyter Notebooks and execute Python code.

Local Anaconda Distribution

- Press the START button, type: **Anaconda Prompt** and run it.
- Go to the desired folder, where you would like to have your project files, using **cd**. In my case: **cd Documents\cloud_computing**
- Type **jupyter-notebook** and press enter.

Great, you now have your library where you can create Jupyter Notebooks and execute Python code.

Create your first Jupyter Notebook file

- Azure:
  - Press: **+ NEW >** word_count.ipynb and select Python 3.6 Notebook. Now press **NEW**
  - **You will see the new file below.**
- Jupyter:
  - Go to **NEW (**top right corner) > select **Python 3.** A new tab will open with the file.
  - In the top left file of the newly opened tab, next to the Jupyter logo you will see **Untitled,** click on it and type in **word_count.**
  - You now have your first jupyter notebook. This will be visible in the first tab, from where you have created it.

Installing MRJob on Azure

In your newly created Library go to **Terminal**, press it and new tab will open. In that new tab, at the command line, type: **pip install mrjob --user** and press enter.

```
nbuser@nbserver:~$ pip install mrjob --user
Collecting mrjob
  Using cached mrjob-0.6.0-py2.py3-none-any.whl
Collecting google-api-python-client>=1.5.0 (from mrjob)
  Using cached google_api_python_client-1.6.4-py2.py3-none-any.whl
Requirement already satisfied: PyYAML>=3.08 in /usr/local/lib/python3.5/dist-packages (from mrjob)
Collecting boto3>=1.4.6 (from mrjob)
  Using cached boto3-1.4.7-py2.py3-none-any.whl
Collecting botocore>=1.6.0 (from mrjob)
  Using cached botocore-1.7.37-py2.py3-none-any.whl
Collecting httplib2<1dev,>=0.9.2 (from google-api-python-client>=1.5.0->mrjob)
Collecting oauth2client<5.0.0dev,>=1.5.0 (from google-api-python-client>=1.5.0->mrjob)
  Using cached oauth2client-4.1.2-py2.py3-none-any.whl
Collecting uritemplate<4dev,>=3.0.0 (from google-api-python-client>=1.5.0->mrjob)
  Using cached uritemplate-3.0.0-py2.py3-none-any.whl
Requirement already satisfied: six<2dev,>=1.6.1 in /usr/local/lib/python3.5/dist-packages (from google-api-python-client>=1.5.0->mrjob)
Collecting s3transfer<0.2.0,>=0.1.10 (from boto3>=1.4.6->mrjob)
  Using cached s3transfer-0.1.11-py2.py3-none-any.whl
Collecting jmespath<1.0.0,>=0.7.1 (from boto3>=1.4.6->mrjob)
  Using cached jmespath-0.9.3-py2.py3-none-any.whl
Collecting docutils>=0.10 (from botocore>=1.6.0->mrjob)
  Using cached docutils-0.14-py3-none-any.whl
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.5/dist-packages (from botocore>=1.6.0->mrjob)
Collecting rsa>=3.1.4 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client>=1.5.0->mrjob)
  Using cached rsa-3.4.2-py2.py3-none-any.whl
Collecting pyasn1>=0.1.7 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client>=1.5.0->mrjob)
  Using cached pyasn1-0.3.7-py2.py3-none-any.whl
Collecting pyasn1-modules>=0.0.5 (from oauth2client<5.0.0dev,>=1.5.0->google-api-python-client>=1.5.0->mrjob)
  Using cached pyasn1_modules-0.1.5-py2.py3-none-any.whl
Installing collected packages: httplib2, pyasn1, rsa, pyasn1-modules, oauth2client, uritemplate, google-api-python-client, docutils, jmespath, b
otocore, s3transfer, boto3, mrjob
Successfully installed boto3-1.4.7 botocore-1.7.37 docutils-0.14 google-api-python-client-1.6.4 httplib2-0.10.3 jmespath-0.9.3 mrjob-0.6.0 oauth
2client-4.1.2 pyasn1-0.3.7 pyasn1-modules-0.1.5 rsa-3.4.2 s3transfer-0.1.11 uritemplate-3.0.0
nbuser@nbserver:~$
```

Installing MRJob locally on Anaconda

Press the START button, type: **Anaconda Prompt** and run it. Now at the command line prompt type: **pip install mrjob**

Anaconda Prompt

```
(C:\Users\stama\Anaconda3) C:\Users\stama>pip install mrjob
Requirement already satisfied: mrjob in c:\users\stama\anaconda3\lib\site-packages
Requirement already satisfied: google-api-python-client>=1.5.0 in c:\users\stama\anaconda3\lib\site-packages (from mrjob
)
Requirement already satisfied: boto3>=1.4.6 in c:\users\stama\anaconda3\lib\site-packages (from mrjob)
Requirement already satisfied: botocore>=1.6.0 in c:\users\stama\anaconda3\lib\site-packages (from mrjob)
Requirement already satisfied: PyYAML>=3.08 in c:\users\stama\anaconda3\lib\site-packages (from mrjob)
Requirement already satisfied: six<2dev,>=1.6.1 in c:\users\stama\anaconda3\lib\site-packages (from google-api-python-cl
ient>=1.5.0->mrjob)
Requirement already satisfied: httplib2<1dev,>=0.9.2 in c:\users\stama\anaconda3\lib\site-packages (from google-api-pyth
on-client>=1.5.0->mrjob)
Requirement already satisfied: uritemplate<4dev,>=3.0.0 in c:\users\stama\anaconda3\lib\site-packages (from google-api-p
ython-client>=1.5.0->mrjob)
```

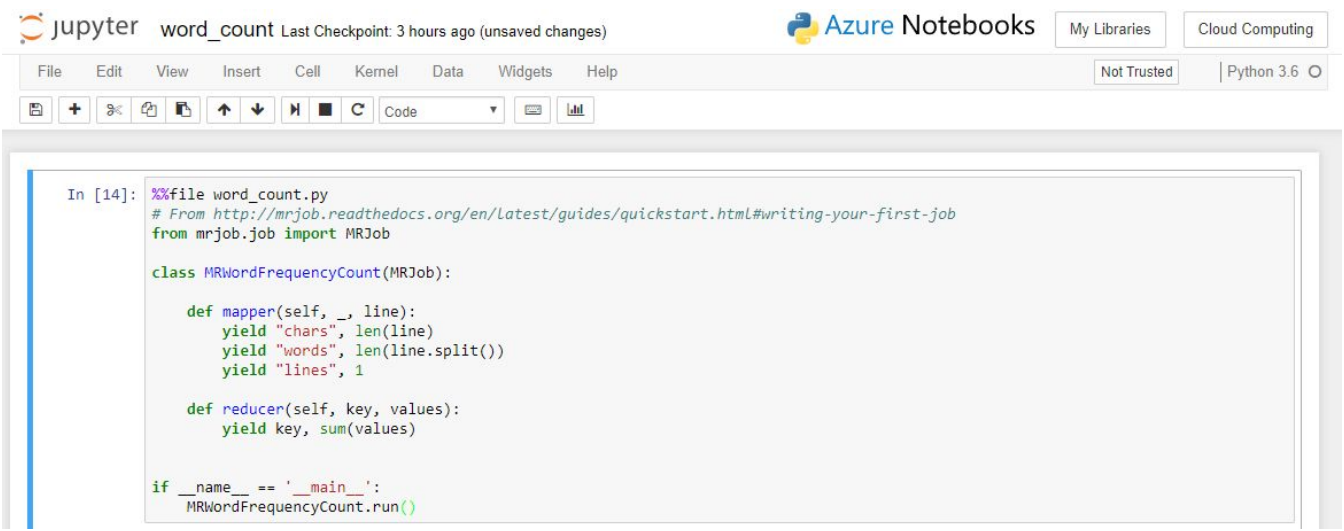
Good job, you can run MapReduce jobs now!

Congrats, this is all you need to run mapreduce jobs locally or in Azure Jupyter notebooks!

Now let’s create and run our first local mapreduce wordcount program

MRJob hello world

Go to your newly created notebooks (word_count.ipynb) and type the code from this link: http://www.dcs.bbk.ac.uk/~cosmin/cc/word_count.ipynb. Go to **Cell > Run Cells.**



Download data to run mapreduce on

Please download the following books in plain text format, which have been sourced from the Gutenberg Project

- http://www.dcs.bbk.ac.uk/~cosmin/cc/data/pg27827.txt
- http://www.dcs.bbk.ac.uk/~cosmin/cc/data/pg3207.txt
- http://www.dcs.bbk.ac.uk/~cosmin/cc/data/pg5200.txt

For Azure you will need to *upload* them to your library,
or *download* them directly using *wget* in the terminal.

Make sure you know the path where you save them as you will need to pass them to your mapreduce program.

Run the MapReduce job

Go back to your word_count notebook, click on the first cell (the one that has all the code inside) and go to: **Insert > Cell Below**. A new cell will be visible below.

Go inside the new cell and type the following:

!python word_count.py -r local *.txt --output-dir=word_count_out --no-output

If you get an error, delete the line and type:

!pip install mrjob

After the installation finishes please try the **!python** line again

That’s it!

If we go back to our notebook dashboard (library in Azure), refresh the page, we will see a new **word_count_out** folder.

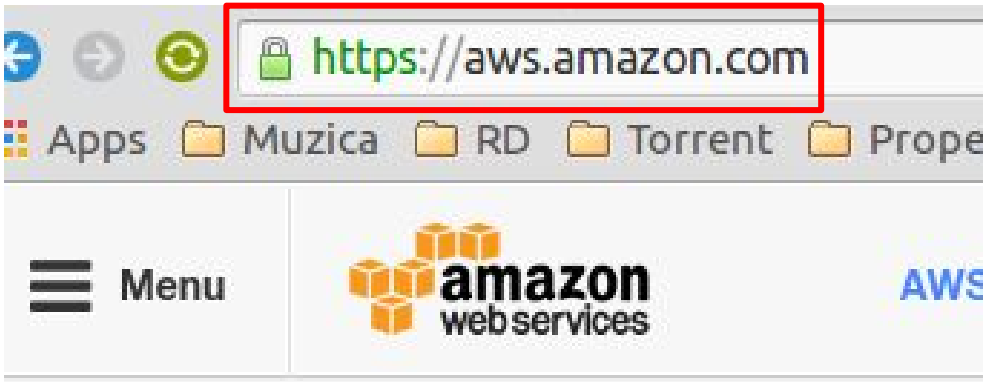
Inside you have the results of your MapReduce script.

Running the file on AWS

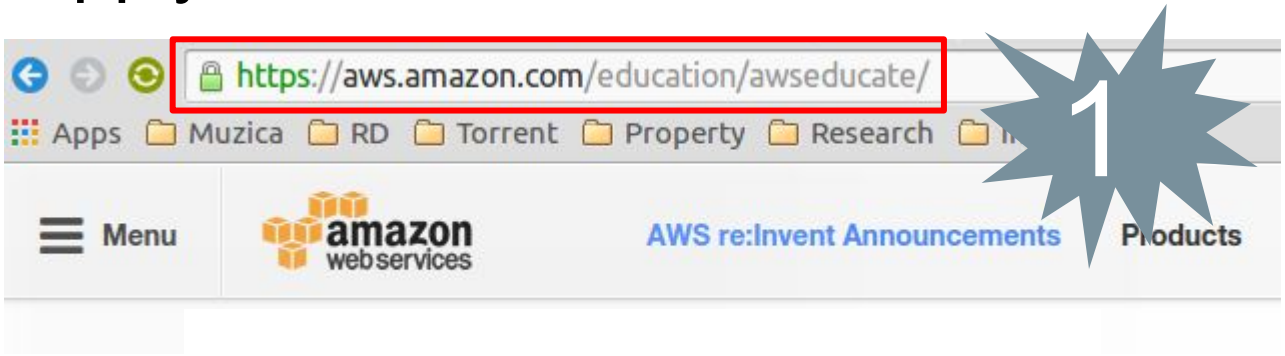
We will need the following:

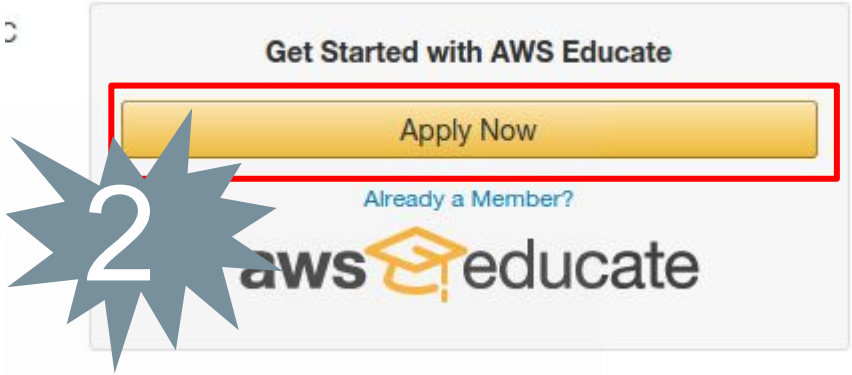
- Create a aws account (and apply the $100 educate credits)
- Create an EC2 Key Pair
- Create S3 storage
  - Upload files to this storage
- Create an MRJob conf file that will automatically create the MapReduce job, execute it and terminate the instances.

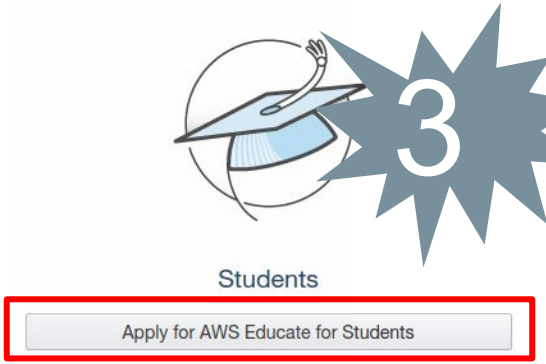
Create a free **AWS** account



Apply for the Academic discount

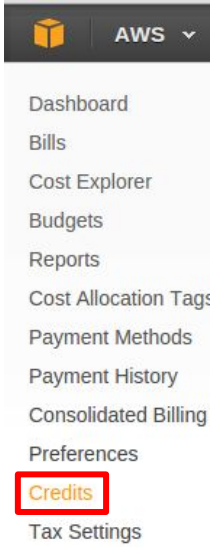


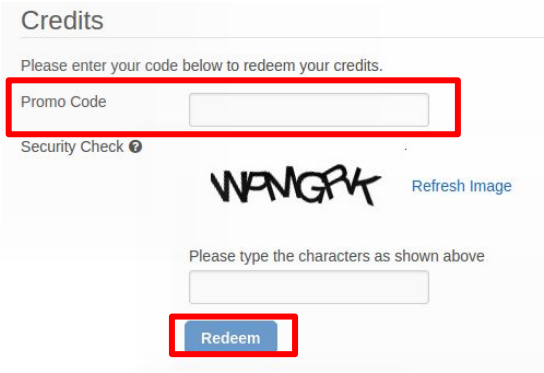




After you receive the **AWS Educate** Application Approved email

- ❑ Go to **My Account > Credits**
- ❑ Paste the promo-code from the approval email and redeem the credits





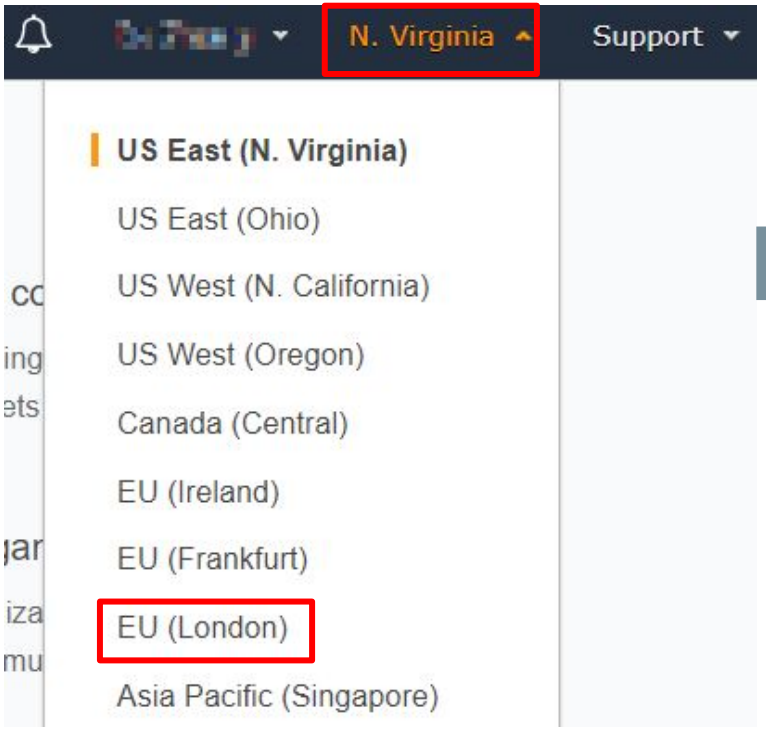
Congratulations, you now have $100 credits!

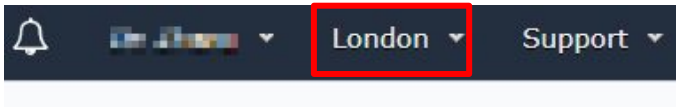
Expiration Date	Credit Name	Credits Used	Credits Remaining
2016-09-30	ENG_FY2015_Q4_100USD	$0.00	$100.00

Total Amount of Credits Remaining: $100.00

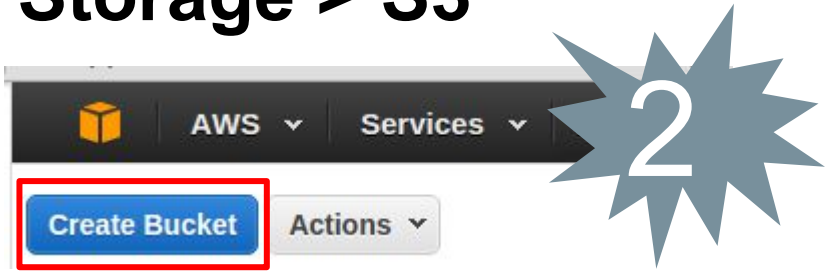
Sign in to the AWS Console

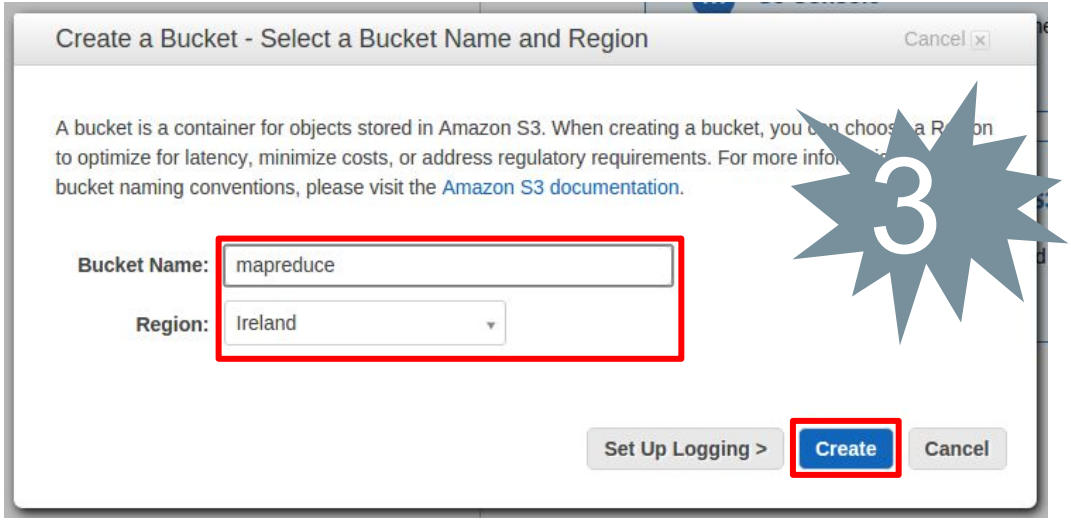
Chose **EU (London)** region



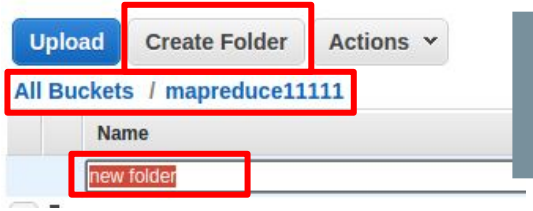


Create a **bucket** under **Storage > S3**

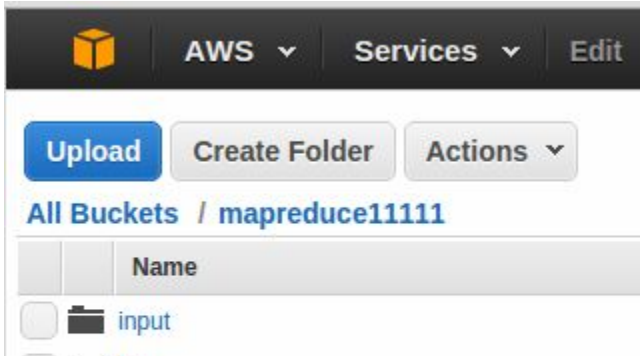




Add folders to the newly created **S3 bucket**



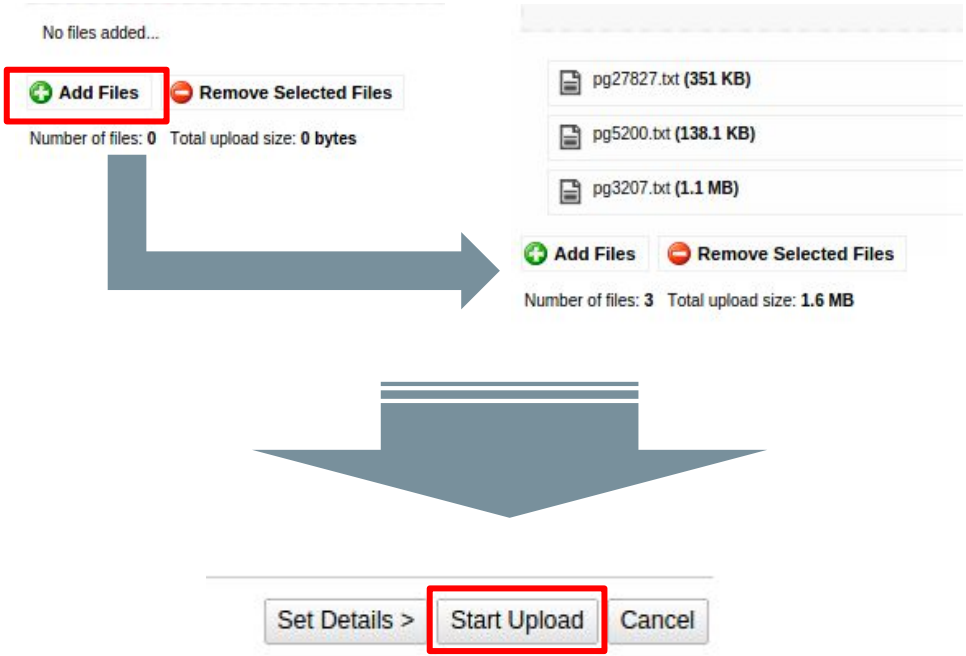
one folder



Upload wordcount books into their **S3 bucket** folder

Select the input folder and press Upload

Add all your input files and upload



Create a new notebook file for the config

In the same way you created the word_count notebook, create a new one with the name **mrjob_conf.ipynb**

It should have the contents from the following link, in its first cell:

http://www.dcs.bbk.ac.uk/~cosmin/cc/mrjob_conf.ipynb

```
[1]: %%file ~/.mrjob.conf
     # http://mrjob.readthedocs.io/en/stable/guides/emr-opts.html

     runners:
       emr:
         aws_access_key_id: AKIAJZ3N2PUZXDBCUOJQ
         aws_secret_access_key: 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
         ec2_key_pair: sec
         ec2_key_pair_file: /home/nbuser/library/sec.pem
         region: eu-west-1 # http://docs.aws.amazon.com/general/latest/gr/rande.html
         master_instance_type: c3.xlarge
         instance_type: c3.xlarge
         num_core_instances: 2
         ssh_tunnel: true

     Writing /home/nbuser/.mrjob.conf
```

Getting all the keys to place in your conf

aws_access_key_id and **aws_secret_access_key**

Go to the AWS Console, click on your name (top right corner) and select My Security Credentials. Click on Continue to Security Credentials if it asks.

Go to Access keys (access key ID and secret access key) and click on Create New Access Key. Press on Show Access Key and Copy and Paste your individual access key in it's appropriate place inside the new mrjob_conf.ipynb tab.

Access Key ID > **aws_access_key_id**

Secret Access Key > **aws_secret_access_key**

Create an EC2 KeyPair

Go to: **Services > EC2**

Select **Key Pairs**, which is under **NETWORK SECURITY** (Left hand side bar).

Press **Create Key Pair** and use any **Key pair name.** Press **Create** after typing name.

Select **Save File,** and save it in **Downloads.**

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair

Get the Key Pair in place

For Azure go to **+ NEW > From Computer > Choose files** and select the downloaded key, it should end in **.pem**

For local Jupyter users make sure you know the location of that file as you will need to update it in the mrjob_conf.ipynb

- **ec2_key_pair**: secure
- **ec2_key_pair_file**: /home/nbuser/library/secure.pem

In my case these are the updates that I need to make.

RUN THE CELL AGAIN!!!

Configuration done!

All done with the MRJob conf, you can now run the cell.

Please have a look at all the configurations that you add to this file:

http://mrjob.readthedocs.io/en/stable/guides/emr-opts.html

You will need to read this if you want to understand what everything does there.

You can control the number of reducers, what types of instances, etc.

```
%%file ~/.mrjob.conf
# http://mrjob.readthedocs.io/en/stable/

runners:
  emr:
    aws_access_key_id: AKIAJZ3N2PUZXDBCU
    aws_secret_access_key: i/gFBUs+ABSwR
    ec2_key_pair: sec
    ec2_key_pair_file: /home/nbuser/libr
    region: eu-west-1 # http://docs.aws.
    master_instance_type: c3.xlarge
    instance_type: c3.xlarge
    num_core_instances: 2
    ssh_tunnel: true
```

```
Overwriting /home/nbuser/.mrjob.conf
```

Running the word_count on AWS EMR

Go back to your word_count.ipynb tab, click on the **!python** cell and select **Insert > Cell below.**

In the newly created cell, type the following:

```
!python word_count.py -r emr s3://mapreduce11111/input/*.txt \
    --output-dir=s3://mapreduce11111/word_count_out \
    --no-output
```

Make sure that you use your S3 bucket name instead of **mapreduce11111.**

Go to **Cell > Run cells.**

Congratulations!

All done, you have successfully ran your first mapreduce program on AWS.

Jupyter notebook tutorial:

https://www.lynda.com/NumPy-tutorials/Introduction-Jupyter-Notebook/508873/543336-4.html

MRJob:

https://pythonhosted.org/mrjob/

Multistep MRJob:

https://www.youtube.com/watch?v=l_wH6cdcRGQ