Objectives

This is a complete documentation on setting up aws server for massive data handling purposes. This tutorial will go over how to create & launch AWS AMI instance and how to attach volume to that instance, along with steps to set up server side of AMI instance.

Step 1: Create and Launch AWS AMI instance

See this link

Step 2: Create & Attach Volume

Volume Creation
(Note: You create up to 16 TB Elastic volume)

Step 3: Connect To Your AMI Instance

```
cd /users/gabriel/directory-that-contains-your-.pem file
ssh -i "AWS.pem" ec2-user@ec2-18-222-26-152.us-east-2.compute.amazonaws.com
```

Step 4: Mounting Volume

Remember you have to mount your volume!!!

```
# mounting a device on aws

# check all volumes and devices
lsblk

# check to see whether there are file system already
sudo file -s /dev/xvdf
# if output "data" then we're good

# make file system
sudo mkfs -t ext4 /dev/xvdf

# create mount point
sudo mkdir /gabriel
```

```
# register
sudo nano /etc/fstab

# at the end of line, add following:
/dev/xvdf   ext4   defaults  0  0

sudo mount -a
df -h
# you will see that the volume is mounted and ready to be used
```

Step 5: Install Anaconda & AWSCLI

```
sudo curl -O https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh
bash Anaconda3-5.0.1-Linux-x86_64.sh
export PATH=~/anaconda3/bin:$PATH
conda update --prefix /home/ec2-user/anaconda3 anaconda
conda install -c conda-forge awscli
```

Step 6: Configure AWSCLI

```
sudo aws configure
Enter your Access Key:
Enter your Secret Key:
None for region
Json for default output file format
```

Step 7: Grab Data From S3

```
sudo aws s3 sync s3://demotaxidata data
# data will be loaded in a directory called data
```

Appendix: Uploading Local File

Simple Put Operation:

```
aws s3 cp train.csv s3://demotaxidata
```

Multipart Upload

```
split -b 250mb train.csv
ws s3api create-multipart-upload --bucket demotaxidata --key train.csv
ws s3api upload-part --bucket demotaxidata --key train.csv --part-number 1 --body xaa
--upload-id
Wo0yaItOy.InFULnVUy9j_Tq37XPQoxNuUfpyvNCQVj_XV.nVfseb1xGdOecRN2YZOk3QR
XgvSFSTL.fhZB51TS1uZtwjAROGedsmoNyMX4-
```

Local Operation

Local

```
--------------------------------------
# install awscli on mac
conda install -c conda-forge awscli

# config
aws configure

sudo awscli configure (as prompted)
cd /user/gabriel/desktop/imgs

# copy data into bucket
# no file size limitation
aws s3 cp train.csv s3://tobaccoimgs
```

Remote

```
---------------------------------------
# Add more RAM (Speed things up a little bit)
https://www.analyticsvidhya.com/blog/2016/05/comprehensive-guide-ml-amazon-web-services-aws/

# Now train data (10.8 GB) is up in the s3
# switch to directory where your AWS.pem(secretkey) file is located
cd /user/gabriel/downloads

# connect to aws sever
ssh -i "AWS.pem" ec2-user@ec2-18-191-151-217.us-east-2.compute.amazonaws.com
```

```
# About how to attach a volume to your AMI instance (if not enough space to accomdate your data)
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-volume.html

# About how to mount your volumes after you attach (I add 16TB...)
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html

# After that...
# change to your new volume
cd /mountdir/

# create data directory that contains your data files
sudo aws s3 sync s3://tobaccoimgs dat

# start training using chunk method to save some memory if needed..


for frame in pd.read_csv('train.csv', parse_dates = ['creation-date', 'null-date'], chunksize = 10 ** 6):
  for algo in Encoders:
     print("Start using %s " % algo.name)
     process_data(frame, isk)
     tmpt = train_algorithm(algo)
     performance_eval(tmpt)

AWS_ACCESS_KEY_ID = 'AKIAJZP3KJXWFCIATXAA'
AWS_SECRET_ACCESS_KEY = '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'
```