AWS PySpark Tutorial
Distributed Data Infrastructures – Fall, 2017

Steps:

1. **Install awscli in your machine.**
   a. Follow the guideline for your operating system here: http://docs.aws.amazon.com/cli/latest/userguide/installing.html
   b. Set your access keys in AWS console > IAM > Security credentials
   c. Set up your access configuration to your local machine. Run "`aws configure`" and fill in your key information.

2. **Set up your S3 bucket.**
   a. Create a new S3 bucket from your AWS console. Make sure you have configured your location.
   b. You can then sync your bucket to your local machine with "aws s3 sync <from> <to>". Example, "`aws s3 sync s3://my-bucket .`" will sync your bucket contents to the working directory.
   c. Upload the data-1-sample.txt file from https://s3.eu-central-1.amazonaws.com/pysparktutorial/data/data-1-sample.txt to your s3 bucket.
   d. You can check all your s3 buckets from your local machine by
   `aws s3 ls`

3. **Set up Elastic Map Reduce (EMR) cluster with spark.**
   a. Go to EMR from your AWS console and Create Cluster.
   b. Fill in cluster name and enable logging. Launch mode should be set to cluster.
   c. EMR release must be 5.7.0 or up.
   d. Select Spark as application type. This will install all required applications for running pyspark.
   e. We will be using a general purpose instance for running spark. Select three instances of type as m4.large.
      i. Every instance has different configuration and costs different. Keep in mind that you will be paying more for larger and more

number of instances if selected. You can find the costs involved from: https://aws.amazon.com/emr/pricing/

f. Select a EC2 key pair. If no key pair has been created, create one from the instructions provided. Save the resulting .pem file in your local computer in a safe location.

g. Roles should be kept as default.

h. Alternatively, you can create the cluster from your console by

```
aws emr create-cluster --name "clustername" --release-
label  emr-5.9.0 --applications Name=Spark Name=Zeppelin
--tags "resource-group-name" --ec2-attributes
KeyName="mykeypair"  --instance-type m4.large --
instance-count 3 --use-default-role
```

where, mykeypair is the name of the EC2 .pem file

4. **Run a spark application**

a. In EMR cluster console, start Zeppelin. Make a new notebook with "spark" as default interpreter

b. Add the following code in the notebook.

```
%pyspark
data = sc.textFile('s3://<path to data-1-sample.txt>')
count = data.count()
print("The count of data set is " + str(count))
```

c. Change the path to data-1-sample.txt file for your S3 bucket.

d. You can learn spark programming at https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html

e. P.S. Zeppelin automatically creates a SparkContext for your program which can be accessed by "sc". Creating another context for your program will give you an error!

5. **Terminate your cluster**! Even acquiring the cluster will build up on your cost. Close it as soon as you are done with the application. It typically takes 2 minutes to close the cluster and 5 minutes to start a new one.

Secondary steps:

1. In order to access the master node to change configurations of installed applications you can SSH by:

   ```
   aws emr ssh --cluster-id <cluster-id> --key-pair-file
   <mykeypair.pem>
   ```

   a. At the master node, you can access a pyspark shell by running command “pyspark”

2. You can also save your zeppelin notebooks directly to your S3 bucket. The following mail in Apache spark mailing list might help: https://mail-archives.apache.org/mod_mbox/incubator-zeppelin-users/201511.mbox/%3CCAF9mLAD1=GW_ghO==Vt1zUCYSoHRrkvaj0N-eyJzt-QEyPBb=A@mail.gmail.com%3E