**Cloud computing cs643 – Project 2**

**JT289 Jinhong Tan**

**April 2021**

**Section1: AWS EMR, 4 instances run spark script**

**Section2: EC2 Instance without Docker**

**Section3: EC2 Instance Docker**

**AWS EMR, 4 instances run spark script**

With Amazon EMR, users can set up a cluster to process and analyze data with big data frameworks. EMR can launch a sample cluster using Spark, and run a simple PySpark script that store in an Amazon S3 bucket. Total 4 instance will create and one is the master instance, other three are workers.

1. Prepare storage for cluster input and output
2. Create an Amazon S3 bucket
3. Upload the PySpark script, trainingdataset.cvs validationdataset.cvs
4. Launch an Amazon EMR cluster
5. Sign into the AWS Management Console and open the Amazon EMR console.
6. Choose **Create cluster** to open the **Quick Options wizard**.
7. Enter a Cluster name, Under Applications, choose the Spark option.
8. Under Security and access choose the EC2 key pair which were created before or create a new one.
9. Click Create cluster to launch the cluster. The cluster should in Waiting status.
10. Submit work to Amazon EMR
11. Click the Cluster you created, make sure it is in a Waiting state
12. Choose Steps, and them choose Add step.
13. For Step type, choose Spark application.
14. For Application location, enter the location of python script in Amazon S3.
15. In the Arguments field. Enter the following arguments and values:

--data\_source s3://jt289project2/trainingdataset.csv

--output\_uri s3://jt289project2/myOutputFolder

1. Choose Add to submit the step. The python script starting to run and will be done after completed.
2. Open S3 console, choose bucket jt289p roject2, choose myoutputfolder. A result file should be stored under this path.

**EC2 Instance without Docker**

1. **Creating EC2 Instance**
2. Under Compute Column in the AWS Management Console Click EC2
3. Under the Instances click Create Instance
4. Select the AMI of your choice. Amazon Linux 2 AMI is usually preferred
5. **St**Select Instance Type I've chosen t2.micro as I'm using AWS Educate and this gives me t2.micro under free tier elgible
6. **Step 5:** Here one can either review and launch or tweak security, configuration and storage features of EC2.
7. Launch EC2 Instance
8. **SSH EC2 instance**
   1. Open terminal, go to the path where pem key pair file saved

Chmod 400 project2key.pem

ssh -i "project2key.pem" root@ec2-54-242-207-60.compute-1.amazonaws.com

* 1. You should have connected to the instance

1. **Installing Spark on EC2 Instance**
   1. Update EC2 in terminal

Sudo yum update -y

* 1. Check python version

python3 –version

* 1. Instal pip

Sudo pip install –upgrade pip

* 1. Install Java

Sudo apt-get install default-jre

Java --version

* 1. Install Py4j used for communicate between java and python

Pip install py4j

* 1. Install Spark and Hadoop

wget http://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop2.7.tgz

sudo tar -zxvf spark-3.0.0-bin-hadoop2.7.tgz

* 1. Install findspark

Sudo pip install findspark

1. **Running your Application in EC2**
   * 1. Upload predict.py file to the Ec2 instance

scp -i <"your .pem file"> predict.py :~/predict.py

* + 1. Run the following command in Ec2 instance to start the model prediction (data from s3 which set to public):

spark-submit --packages org.apache.hadoop:hadoop-aws:2.7.4 predict.py s3://jt289project2/ValidationDataset.csv

**EC2 Instance with Docker**

Following the procedures from above to create instance to set up the environment. (step 1 to step7).

1. **Installation**
2. Install the Docker package

sudo yum install docker -y

1. Run the Docker service

sudo service docker start

1. Add EC2 user to the docker group

sudo usermod -a -G docker ec2-user

1. Verify the EC2-user can run Docker commands

docker --version or docker info

1. **Create a Dockerfile**

touch Dockerfile

1. nano Dockerfile and create the Dockerfile Image to automate the process

sudo docker build . -f Dockerfile -t <Image name of your choice>

Pushing and Pulling created Image to DockerHub

1. Login to your dockerhub account through ec2

docker login: Type your credentials

1. In order to push docker type the following commands

docker tag <Local Ec2 Repository name>:<Tag name> <dockerhub username>/<local Ec2 Repository name>

docker push <dockerhub username>/<local Ec2 Repository name>

1. **Pulling your Dockerimage back to Ec2**

docker pull <dockerhub username>/<Repository name>:<tag name>

Example:

docker pull sampathgonnuru/cs643-project2:latest

1. Running my dockerimage

sudo docker run -t <Given Image name>

docker run -it sampathgonnuru/cs643-project2:latest s3//mywineproject/ValidationDataset.csv