

CS643 Programming Assignment 2

Name : Pavan Sagar T

Docker hub link : <https://hub.docker.com/repository/docker/psagar12/assignment2/general>

Objectives

- to use Apache Spark to train an ML model in parallel on multiple EC2 instances
- to use Spark's MLlib to develop and use an ML model in the cloud
- to use Docker to create a container for your ML model to simplify model deployment

Setup

First we have to create EMR cluster using AWS management console
We should have 4 EC2 instances and using Spark for the ML application

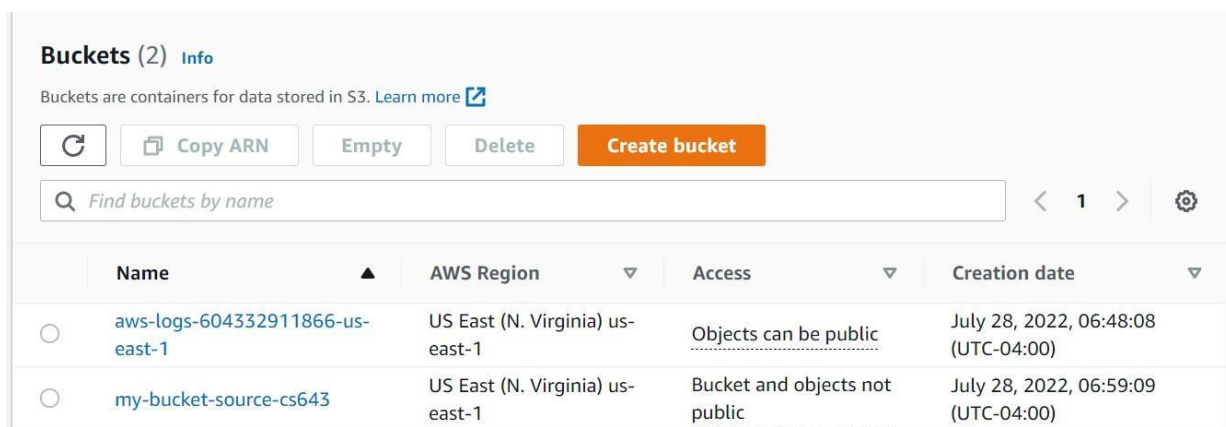
In the 4 Ec2 instances we have one master node and 3 core nodes

Master: Running 1 m5.xlarge

Core: Running 3 m5.xlarge

After setting up our cluster we need to create S3 bucket which acts as storage and helps to fetch the required data faster and send output to that bucket

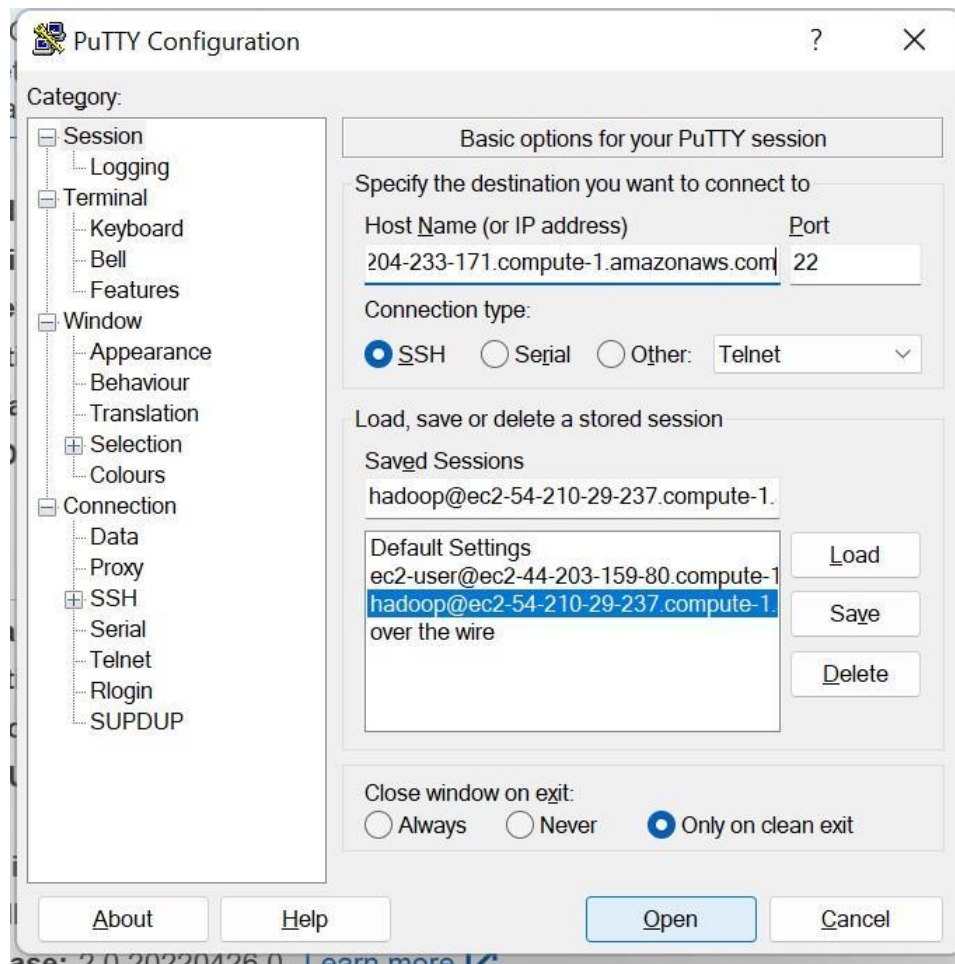
One bucket is created for the logs of the cluster and the other for storing our data



Buckets (2) Info				
Buckets are containers for data stored in S3. Learn more				
	Copy ARN	Empty	Delete	Create bucket
<input type="text" value="Find buckets by name"/>				
	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	aws-logs-604332911866-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	July 28, 2022, 06:48:08 (UTC-04:00)
<input type="radio"/>	my-bucket-source-cs643	US East (N. Virginia) us-east-1	Bucket and objects not public	July 28, 2022, 06:59:09 (UTC-04:00)

Model training and Application Prediction

Now connect to the master instance using any SSH client . I am using Putty to connect to the master node. Then using WinSCP copy the spark application to run on the cluster



In the home directory i have my training.py file which i am using to train my model using the TrainingDataset.csv

By using the command 'spark-submit training.py' we can run the application on multiple clusters with the help of serialize

The s3 bucket URI is given in the document to fetch the files from the S3 bucket and for saving the model

I have also used ValidationDataset.csv to adjust my hyperparameters and tune it to highest score possible

Now we have to create Docker container and run our application on it using the docker file created

We use following commands to run docker container using the docker image

```
docker build -t docker-ml-model -f Dockerfile .
```

```
docker run docker-ml-model
```

After creating our docker successfully we can run our prediction test