Programming Assignment 2

Name: Utakarsh Aggarwal

Class: CS643-852

Docker Hub Link:

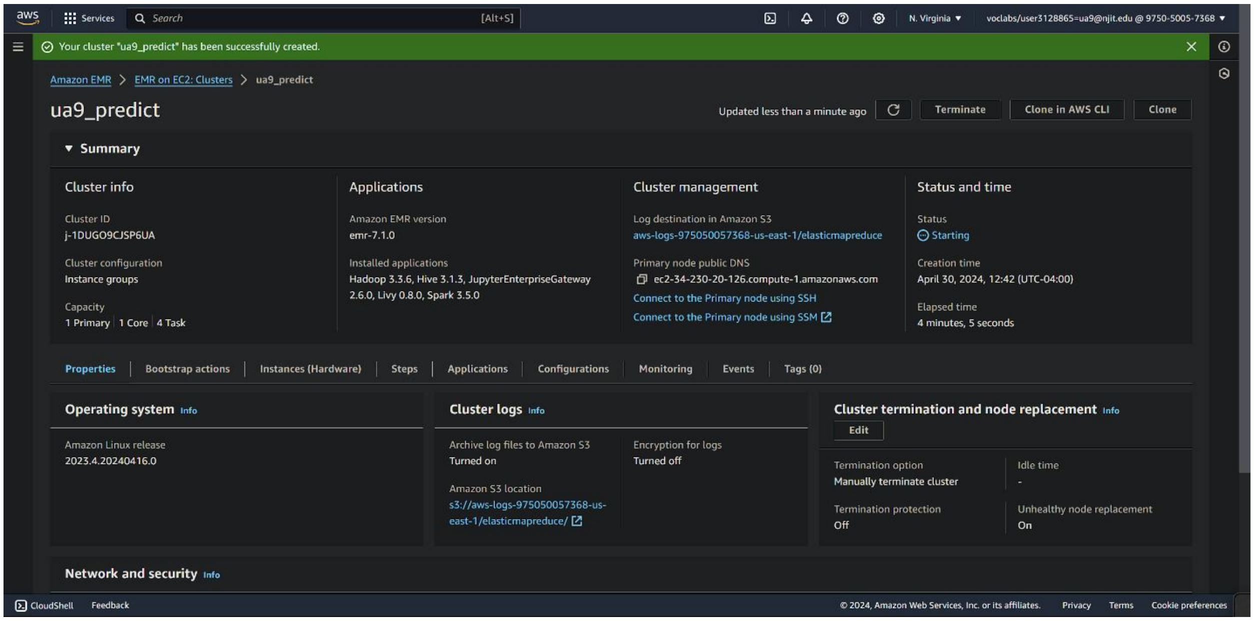
https://hub.docker.com/repository/docker/utakarshagg/winequalitytesting/general

Github Link:

https://github.com/UtakarshAgg/Wine-Quality-Testing---Programming-Assignment-2

Instructions:

1. Launch AWS Academy Learner Lab
2. Create EMR cluster
   2.1. Add up to 4 tasks inside the cluster under *Cluster Configuration*
   2.2. Create or upload a keypair (.ppk) under *Security Configuration*
   2.3. Change IAM roles to default roles under *Access Management*



Your cluster "ua9_predict" has been successfully created.

ua9_predict

Summary

Cluster info

Cluster ID
j-1DUGO9CJSP6UA

Cluster configuration
Instance groups

Capacity
1 Primary | 1 Core | 4 Task

Applications

Amazon EMR version
emr-7.1.0

Installed applications
Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.0

Cluster management

Log destination in Amazon S3
aws-logs-975050057368-us-east-1/elasticmapreduce

Primary node public DNS
ec2-34-230-20-126.compute-1.amazonaws.com
Connect to the Primary node using SSH
Connect to the Primary node using SSM

Status and time

Status
Starting

Creation time
April 30, 2024, 12:42 (UTC-04:00)

Elapsed time
4 minutes, 5 seconds

Operating system Info

Amazon Linux release
2023.4.20240416.0

Cluster logs Info

Archive log files to Amazon S3
Turned on

Encryption for logs
Turned off

Amazon S3 location
s3://aws-logs-975050057368-us-east-1/elasticmapreduce/

Cluster termination and node replacement Info

Termination option
Manually terminate cluster

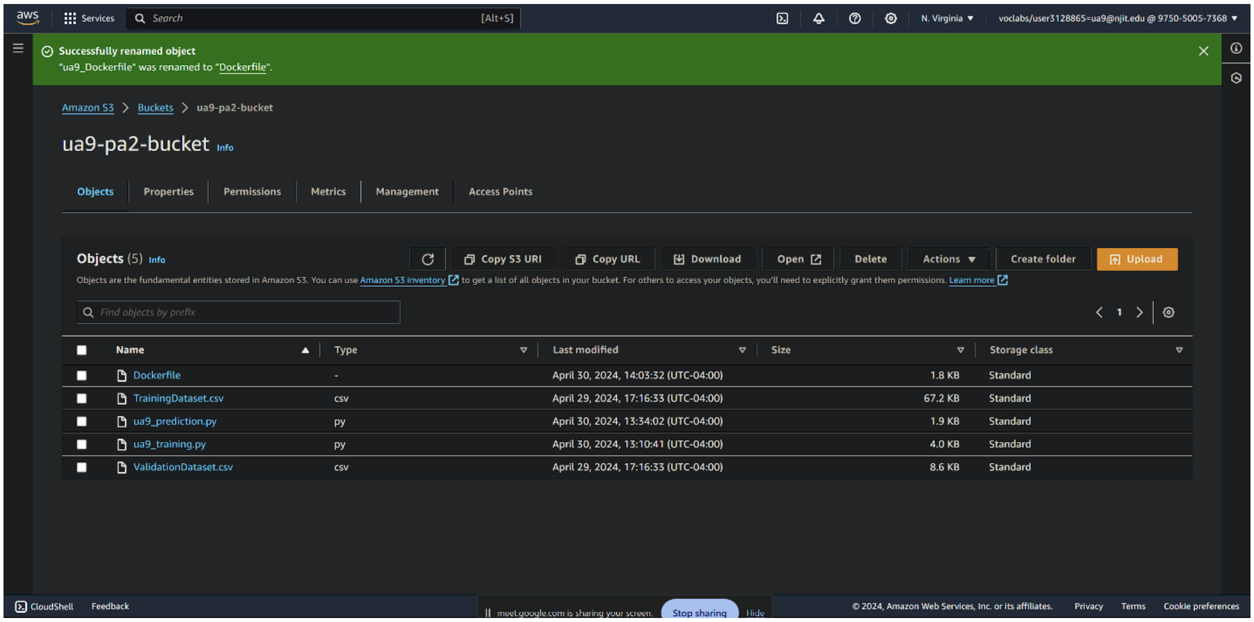
Idle time
-

Termination protection
Off

Unhealthy node replacement
On

Network and security Info

3. Create S3 Bucket
    3.1. No configuration change required
4. Upload all required files in S3 bucket
    4.1. Provided datasets
    4.2. Training and prediction code
    4.3. Docker file



Successfully renamed object
"ua9_Dockerfile" was renamed to "Dockerfile".

Amazon S3 > Buckets > ua9-pa2-bucket

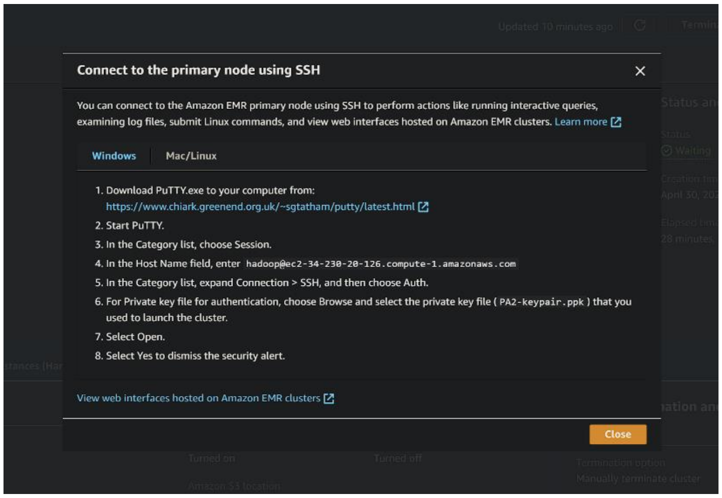
ua9-pa2-bucket Info

Objects (5) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
Dockerfile	-	April 30, 2024, 14:03:32 (UTC-04:00)	1.8 KB	Standard
TrainingDataset.csv	csv	April 29, 2024, 17:16:33 (UTC-04:00)	67.2 KB	Standard
ua9_prediction.py	py	April 30, 2024, 13:34:02 (UTC-04:00)	1.9 KB	Standard
ua9_training.py	py	April 30, 2024, 13:10:41 (UTC-04:00)	4.0 KB	Standard
ValidationDataset.csv	csv	April 29, 2024, 17:16:33 (UTC-04:00)	8.6 KB	Standard

5. Open the EMR cluster
    5.1. Find 'Connect to primary node using SSH'
    5.2. Copy the hostname mentioned under Windows tab



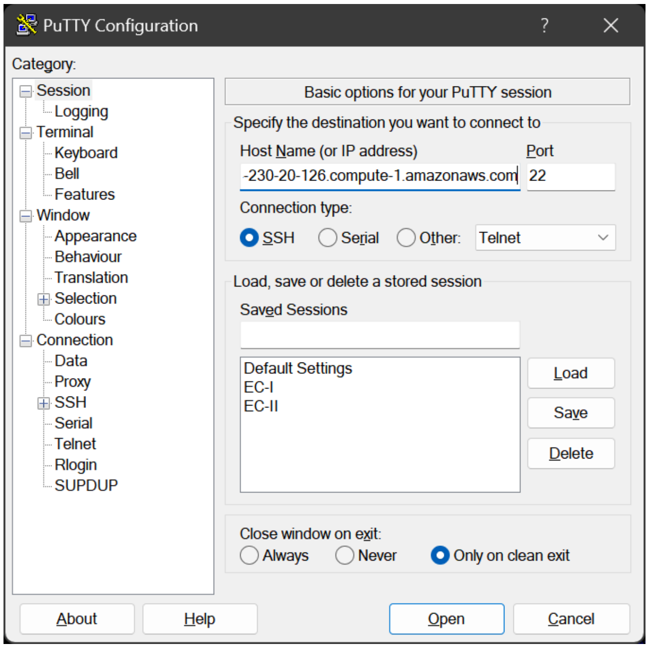
Connect to the primary node using SSH

You can connect to the Amazon EMR primary node using SSH to perform actions like running interactive queries, examining log files, submit Linux commands, and view web interfaces hosted on Amazon EMR clusters. Learn more

1. Download PuTTY.exe to your computer from:
   https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
2. Start PuTTY.
3. In the Category list, choose Session.
4. In the Host Name field, enter `hadoop@ec2-34-230-20-126.compute-1.amazonaws.com`
5. In the Category list, expand Connection > SSH, and then choose Auth.
6. For Private key file for authentication, choose Browse and select the private key file (`PA2-keypair.ppk`) that you used to launch the cluster.
7. Select Open.
8. Select Yes to dismiss the security alert.

View web interfaces hosted on Amazon EMR clusters

6. Open PuTTy
    6.1. Paste the copied hostname under *Session*
    6.2. Go to Connection >> SSH >> Auth >> Credentials
    6.3. Upload the keypair (.ppk)
    6.4. Launch the session



Specify the destination you want to connect to

Host Name (or IP address): -230-20-126.compute-1.amazonaws.com

Port: 22

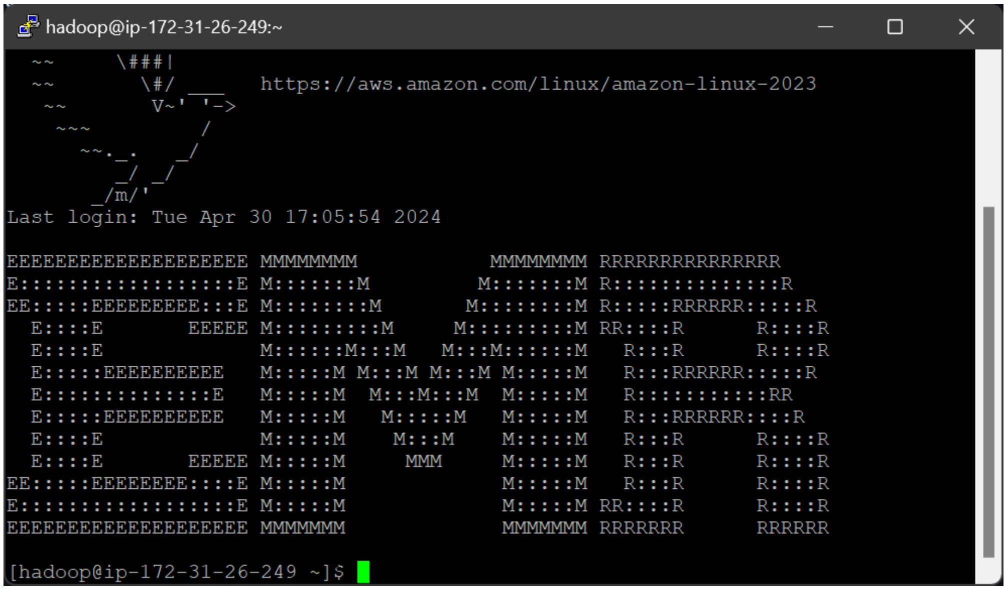
Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings
EC-I
EC-II

Close window on exit: ○ Always ○ Never ● Only on clean exit



```
  ~~       \###|
  ~~         \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~         V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Apr 30 17:05:54 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-26-249 ~]$
```

7. Run the commands below in the EMR session

    7.1. Switch to root user
    - sudo su

    7.2. Install Numpy
    - pip install numpy

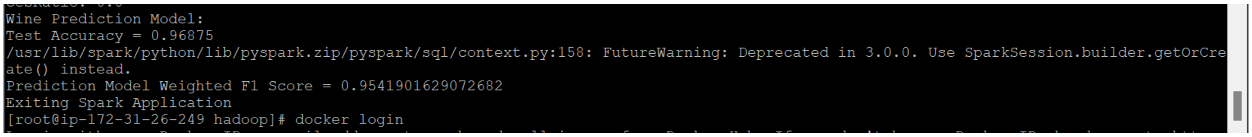
    7.3. Sync all files in the instance
    - aws s3 sync s3://ua9-pa2-bucket/ .
    - You can verify the synced files using ‘ls’ command

    7.4. Run the training script
    - spark-submit ua9_training.py

    7.5. Copy the files in Hadoop File System
    - hadoop fs -copyFromLocal TrainingDataset.csv hdfs://ip-172-31-26-249.ec2.internal:8820/user/root/
    - hadoop fs -copyFromLocal ValidationDataset.csv hdfs://ip-172-31-26-249.ec2.internal:8820/user/root/

    7.6. Copy the model from HDFS to root directory
    - hadoop fs -ls hdfs://ip-172-31-26-249.ec2.internal:8820/user/root/
    - hadoop fs -get hdfs://ip-172-31-26-249.ec2.internal:8820/user/root/ua9-trainedmodel .
    - Check imported model in root directory using ’ls’ command

    7.7. Run the model for prediction
    - spark-submit ua9_prediction.py
    - The test score and weighted F1 score is printed



```
Wine Prediction Model:
Test Accuracy = 0.96875
/usr/lib/spark/python/lib/pyspark.zip/pyspark/sql/context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCre
ate() instead.
Prediction Model Weighted F1 Score = 0.9541901629072682
Exiting Spark Application
[root@ip-172-31-26-249 hadoop]# docker login
```

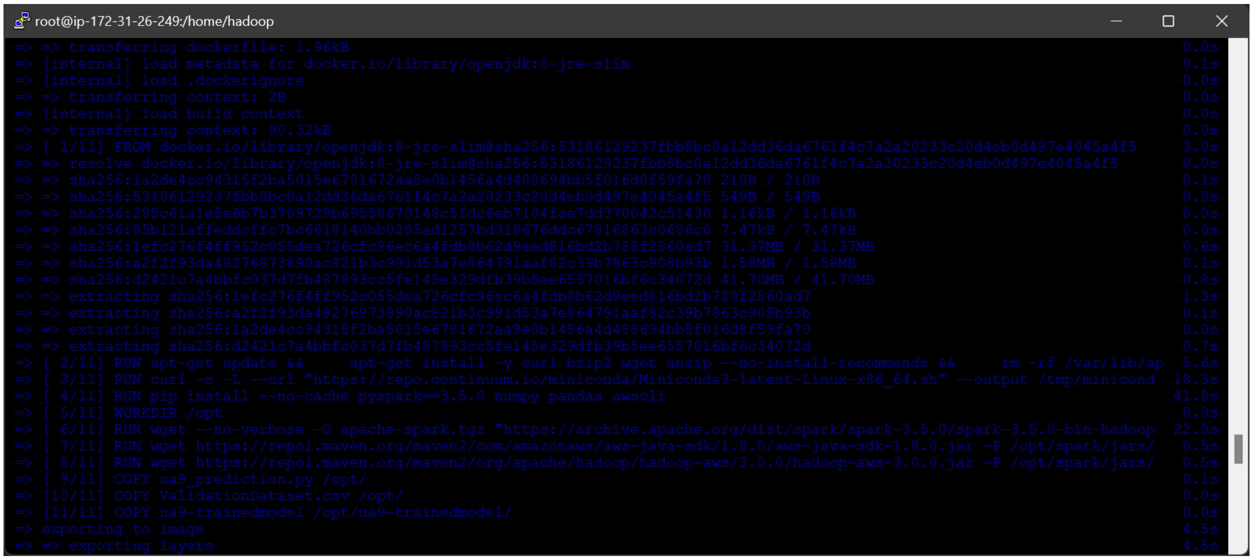
8. Running the prediction model using Docker

8.1. Login into docker

- docker login

8.2. Create the docker image using dockerfile

- docker build -t utakarshagg/winequalitytesting .



```
=> => transferring dockerfile: 1.96kB                                                                                            0.0s
=> [internal] load metadata for docker.io/library/openjdk:8-jre-slim                                                             0.1s
=> [internal] load .dockerignore                                                                                                 0.0s
=> => transferring context: 2B                                                                                                   0.0s
=> [internal] load build context                                                                                                 0.0s
=> => transferring context: 90.32kB                                                                                              0.0s
=> [ 1/11] FROM docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5   3.0s
=> => resolve docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5     0.0s
=> => sha256:1a2de4cc94315f2ba5015e6781672aa8e0b1456a4d488694bb5f016d8f59fa70 210B / 210B                                        0.1s
=> => sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5 549B / 549B                                        0.0s
=> => sha256:285c61a1e5e6b7b3709729b69558670148c5fdc6eb7104fae7dd370042c51430 1.16kB / 1.16kB                                    0.0s
=> => sha256:85b121affeddcffc7bc6618140bb0285ad1257bd318676ddc67816863c0686c0 7.47kB / 7.47kB                                    0.0s
=> => sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7 31.37MB / 31.37MB                                  0.6s
=> => sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b 1.58MB / 1.58MB                                    0.1s
=> => sha256:d2421c7a4bbfc037d7fb487893cc5fe145e329dfb39b5ee6557016bf6c34072d 41.70MB / 41.70MB                                  0.8s
=> => extracting sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7                                         1.3s
=> => extracting sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b                                         0.1s
=> => extracting sha256:1a2de4cc94315f2ba5015e6781672aa8e0b1456a4d488694bb5f016d8f59fa70                                         0.0s
=> => extracting sha256:d2421c7a4bbfc037d7fb487893cc5fe145e329dfb39b5ee6557016bf6c34072d                                         0.7s
=> [ 2/11] RUN apt-get update &&     apt-get install -y curl bzip2 wget unzip --no-install-recommends &&     rm -rf /var/lib/ap  5.6s
=> [ 3/11] RUN curl -s -L --url "https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh" --output /tmp/minicond  18.3s
=> [ 4/11] RUN pip install --no-cache pyspark==3.5.0 numpy pandas awscli                                                        41.8s
=> [ 5/11] WORKDIR /opt                                                                                                          0.0s
=> [ 6/11] RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop  22.0s
=> [ 7/11] RUN wget https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk/1.8.0/aws-java-sdk-1.8.0.jar -P /opt/spark/jars/   0.5s
=> [ 8/11] RUN wget https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-aws/3.0.0/hadoop-aws-3.0.0.jar -P /opt/spark/jars/   0.5s
=> [ 9/11] COPY ua9_prediction.py /opt/                                                                                          0.1s
=> [10/11] COPY ValidationDataset.csv /opt/                                                                                      0.0s
=> [11/11] COPY ua9-trainedmodel /opt/ua9-trainedmodel/                                                                          0.0s
=> exporting to image                                                                                                            4.5s
=> => exporting layers                                                                                                           4.5s
```

8.3. Run the image

- docker run utakarshagg/winequalitytesting

8.4. Push the image in docker hub

- docker push utakarshagg/winequalitytesting