CS 643, Cloud Computing – Programming Assignment 2

Mehul Shah
ms298@njit.edu

Weblinks:

- GITHUB Repository: https://github.com/ms298njit/CS643-PA2-WQP
- Docker Container: https://hub.docker.com/repository/docker/ms298/wqp

Purpose:

The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications in Amazon AWS cloud platform. Specifically:

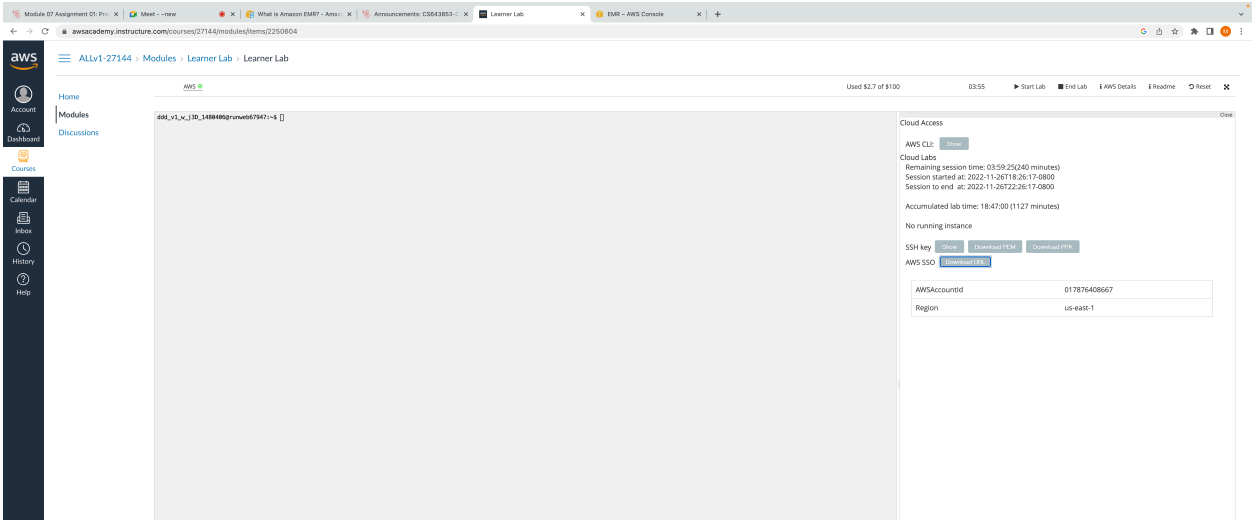
- how to use Apache Spark to train an ML model in parallel on multiple EC2 instances
- how to use Spark's MLlib to develop and use an ML model in the cloud
- how to use Docker to create a container for your ML model to simplify model deployment

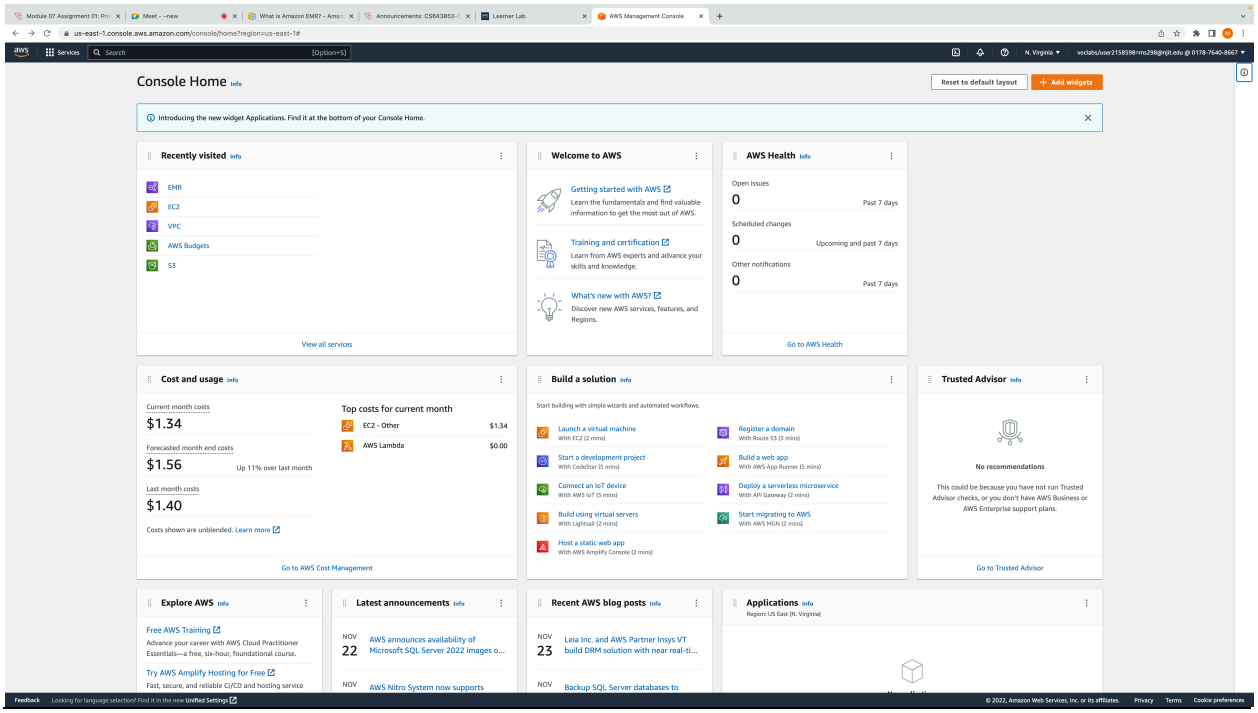
Description:

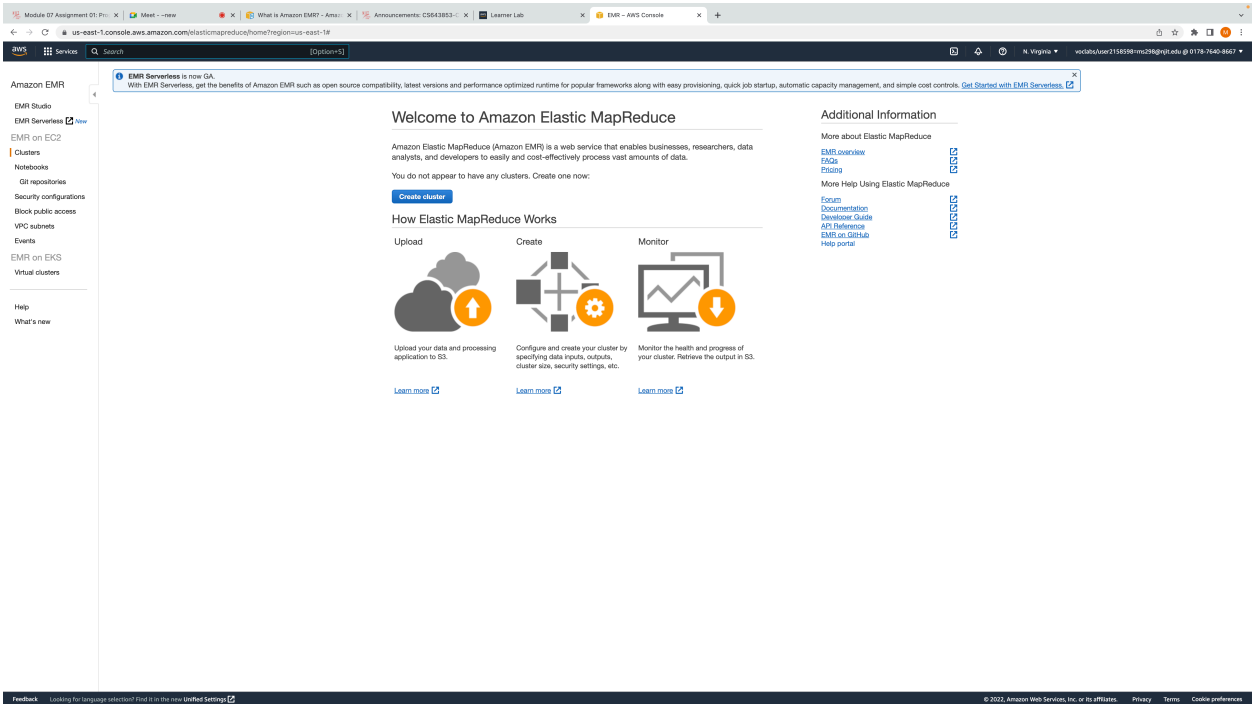
Build a wine quality prediction ML model in Spark over AWS. The model must be trained in parallel using 4 EC2 instances. Then, save and load the model in a Spark application that will perform wine quality prediction; this application will run on one EC2 instance. The assignment must be implemented in Java.

Instructions:

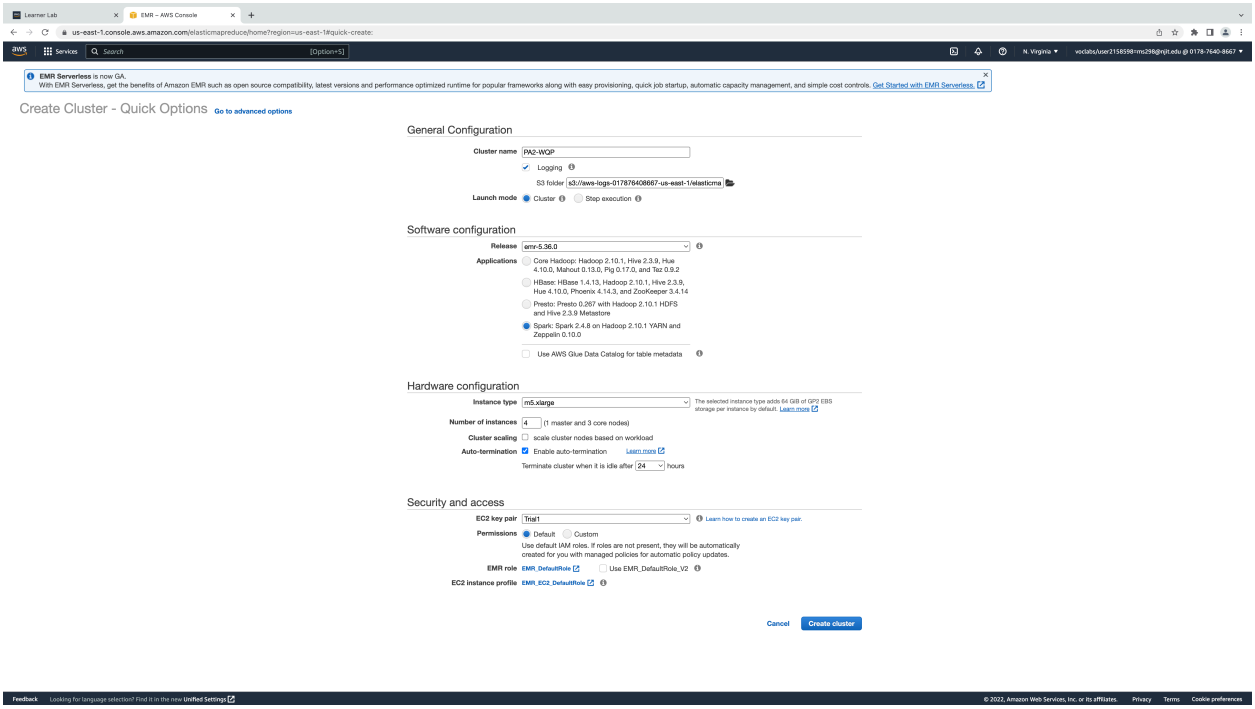
1. Create spark cluster using EMR AWS. Log on to AWS academy student account, start learner lab and open EMR-AWS console.

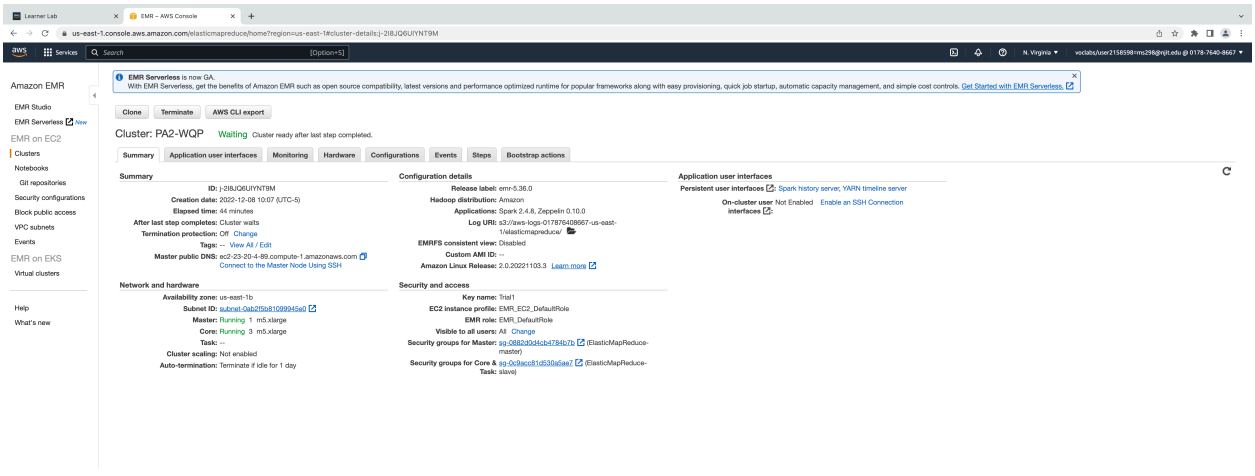


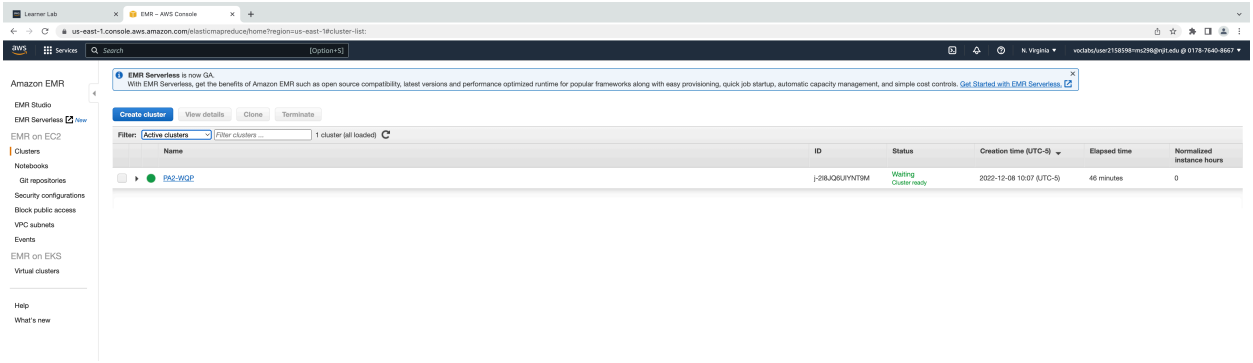

Console Home Info
Reset to default layout
Add widgets
Introducing the new widget Applications. Find it at the bottom of your Console Home.
Recently visited Info
EMR
EC2
VPC
AWS Budgets
S3
View all services
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.
AWS Health Info
Open issues
0
Past 7 days
Scheduled changes
0
Upcoming and past 7 days
Other notifications
0
Past 7 days
Go to AWS Health
Cost and usage Info
Current month costs
$1.34
Forecasted month end costs
$1.56
Up 11% over last month
Last month costs
$1.40
Costs shown are unblended. Learn more
Top costs for current month
EC2 - Other $1.34
AWS Lambda $0.00
Go to AWS Cost Management
Build a solution Info
Start building with simple wizards and automated workflows.
Launch a virtual machine
With EC2 (2 mins)
Register a domain
With Route 53 (5 mins)
Start a development project
With CodeStar (5 mins)
Build a web app
With AWS App Runner (5 mins)
Connect an IoT device
With AWS IoT (5 mins)
Deploy a serverless microservice
With API Gateway (2 mins)
Build using virtual servers
With Lightsail (2 mins)
Start migrating to AWS
With AWS MGN (2 mins)
Host a static web app
With AWS Amplify Console (2 mins)
Trusted Advisor Info
No recommendations
This could be because you have not run Trusted Advisor checks, or you don't have AWS Business or AWS Enterprise support plans.
Go to Trusted Advisor
Explore AWS Info
Free AWS Training
Advance your career with AWS Cloud Practitioner Essentials—a free, six-hour, foundational course.
Try AWS Amplify Hosting for Free
Fast, secure, and reliable CI/CD and hosting service
Latest announcements Info
NOV 22 AWS announces availability of Microsoft SQL Server 2022 images o...
NOV AWS Nitro System now supports
Recent AWS blog posts Info
NOV 23 Leia Inc. and AWS Partner Insys VT build DRM solution with near real-ti...
NOV Backup SQL Server databases to
Applications Info
Region: US East (N. Virginia)
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Amazon EMR
EMR Studio
EMR Serverless New
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
Welcome to Amazon Elastic MapReduce
Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.
You do not appear to have any clusters. Create one now:
Create cluster
How Elastic MapReduce Works
Upload
Upload your data and processing application to S3.
Learn more
Create
Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.
Learn more
Monitor
Monitor the health and progress of your cluster. Retrieve the output in S3.
Learn more
Additional Information
More about Elastic MapReduce
EMR overview
FAQs
Pricing
More Help Using Elastic MapReduce
Forum
Documentation
Developer Guide
API Reference
EMR on GitHub
Help portal
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

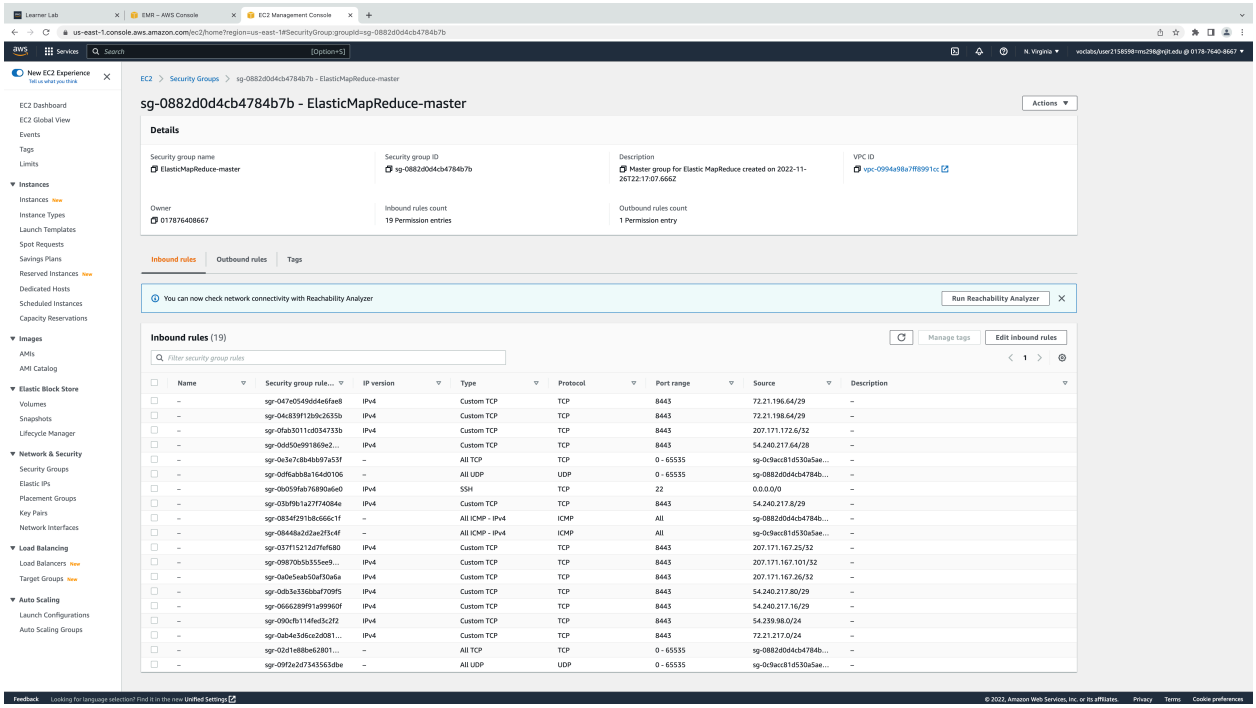
2. Configure Cluster parameters such as: General, Software, Hardware, Security and Access

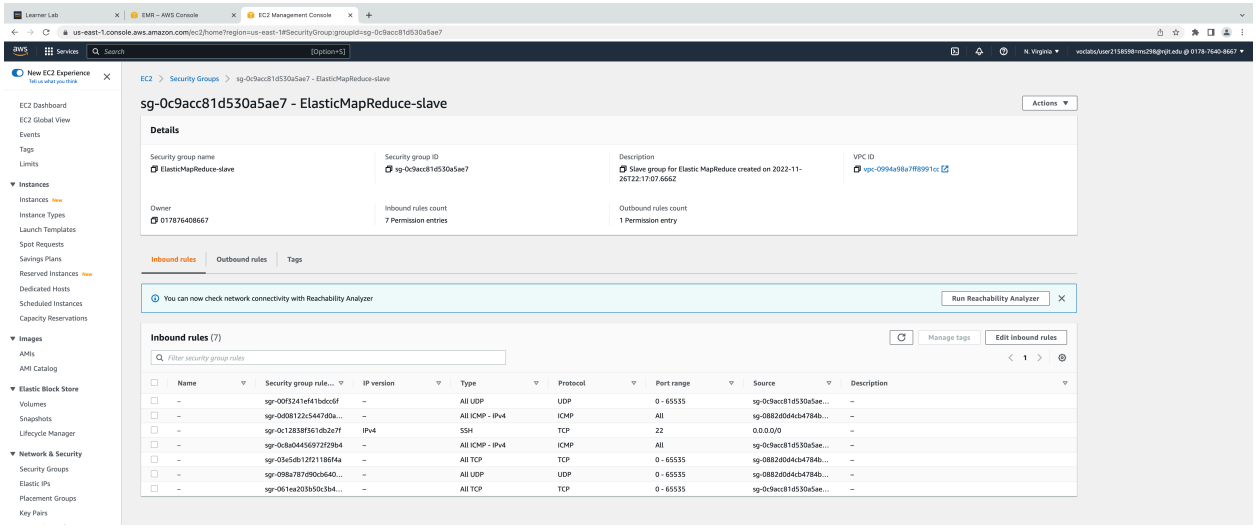






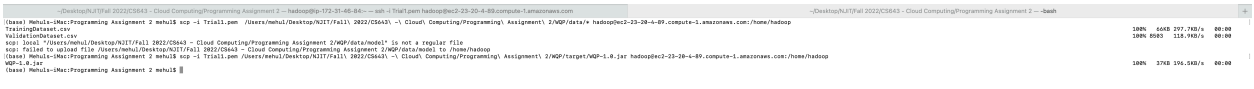
3. Edit inbound rules to allow access the cluster through SSH

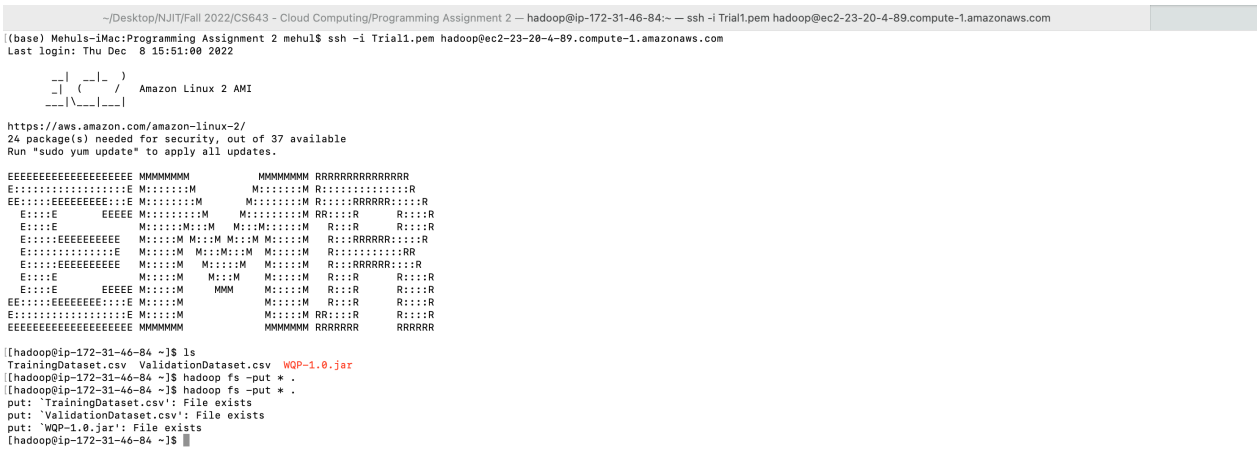




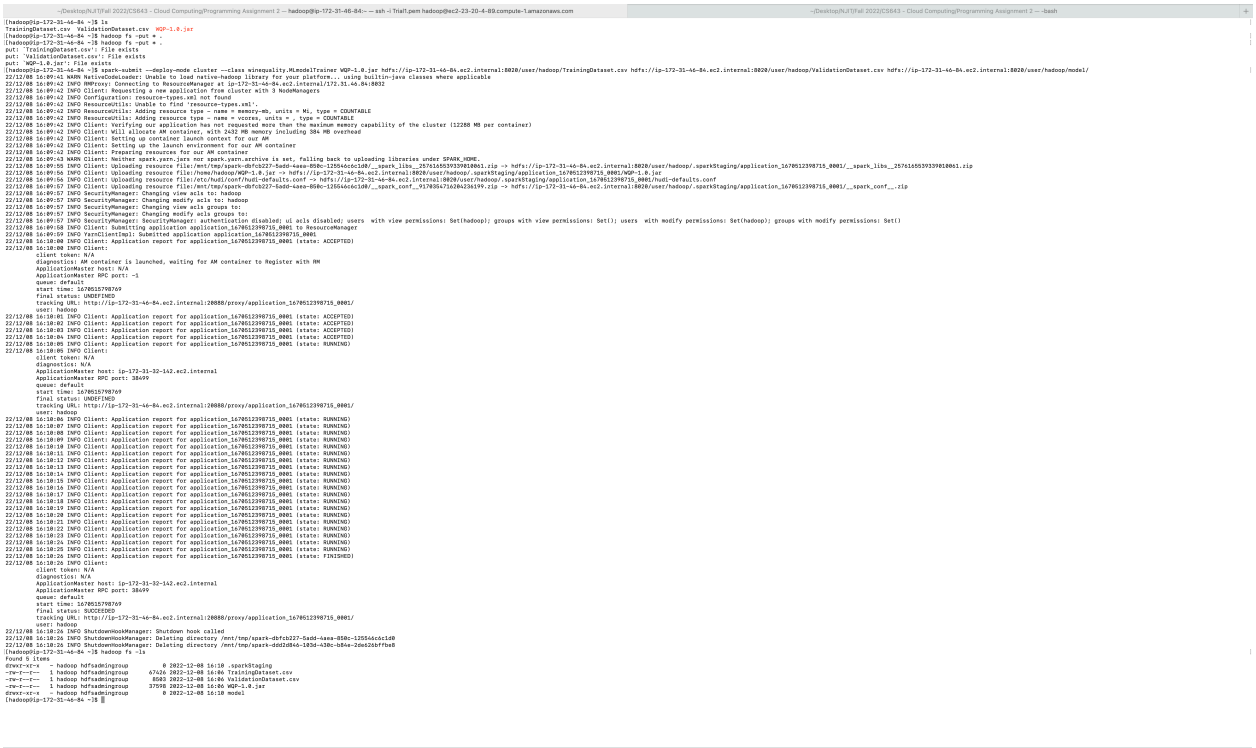
4. Connect to master node/instance within cluster using command:
   - ssh -i Trail1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com

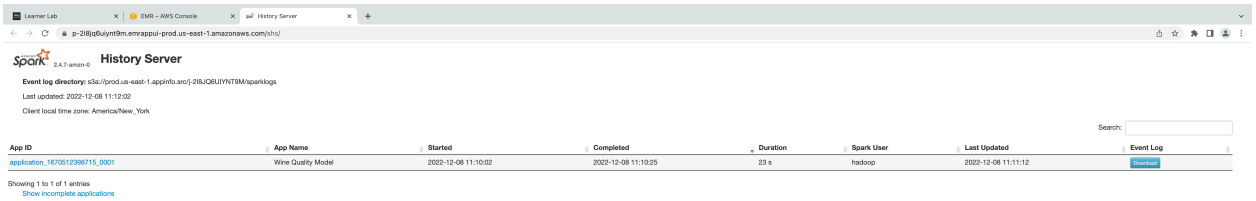
5. copy files (TrainingDataset.csv, ValidationDataset.csv and WQP-1.0.jar) to EMR local file system and then put it on HDFS
   - scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall\ 2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/data/* hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
   - scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall\ 2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/target/WQP-1.0.jar hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
   - hadoop fs -put * .

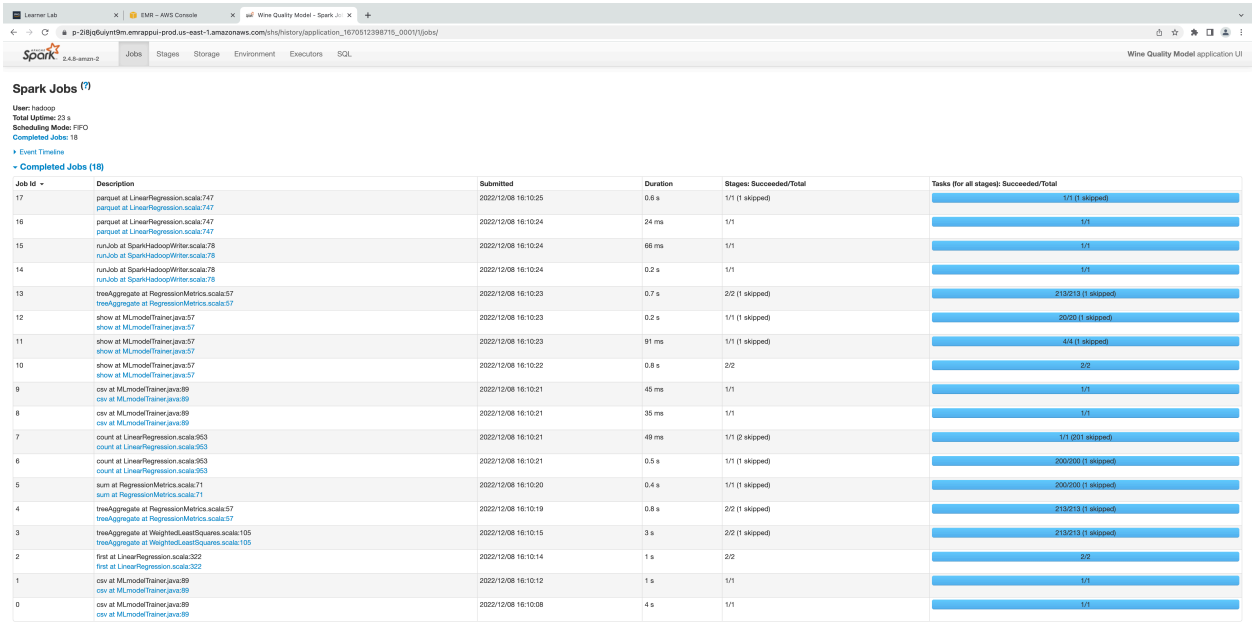




6. Create and run spark application to train ML model in parallel on 4 EC2 instances.
   - Read TrainingDataset.csv, ValidationDataset.csv from HDFS.
   - Create trained ML model onto HDFS.
   - Run WQP-1.0.Jar file with three arguments: 1st Argument = TrainingDataset.csv, 2nd Argument = ValidationDataset.csv, 3rd Argument = trained ML model path
   - spark-submit --deploy-mode cluster --class winequality.MLmodelTrainer WQP-1.0.jar hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/TrainingDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/model/
   - hadoop fs -ls







Learner Lab | EMR – AWS Console | Wine Quality Model - Spark Jo

p-2i8jq6uiynt9m.emrappui-prod.us-east-1.amazonaws.com/shs/history/application_1670512398715_0001/1/jobs/

Spark 2.4.8-amzn-2 — Jobs | Stages | Storage | Environment | Executors | SQL — Wine Quality Model application UI

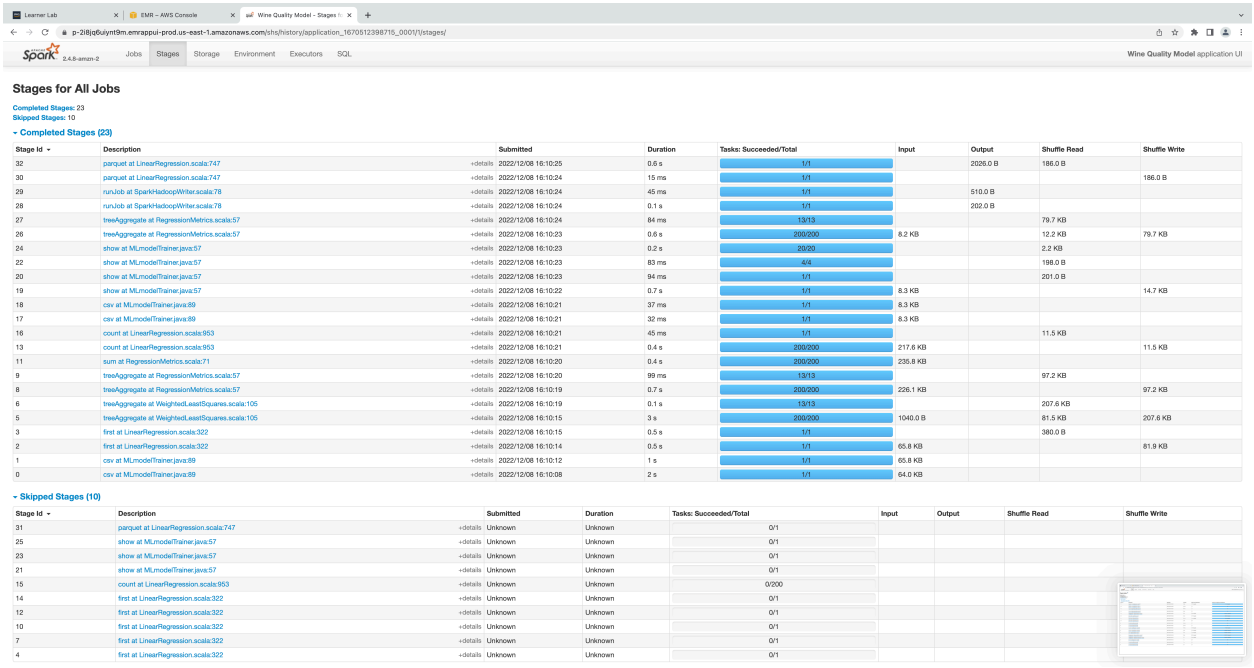
Spark Jobs (?)

User: hadoop
Total Uptime: 23 s
Scheduling Mode: FIFO
Completed Jobs: 18

▸ Event Timeline

▾ Completed Jobs (18)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
17	parquet at LinearRegression.scala:747 parquet at LinearRegression.scala:747	2022/12/08 16:10:25	0.6 s	1/1 (1 skipped)	1/1 (1 skipped)
16	parquet at LinearRegression.scala:747 parquet at LinearRegression.scala:747	2022/12/08 16:10:24	24 ms	1/1	1/1
15	runJob at SparkHadoopWriter.scala:78 runJob at SparkHadoopWriter.scala:78	2022/12/08 16:10:24	66 ms	1/1	1/1
14	runJob at SparkHadoopWriter.scala:78 runJob at SparkHadoopWriter.scala:78	2022/12/08 16:10:24	0.2 s	1/1	1/1
13	treeAggregate at RegressionMetrics.scala:57 treeAggregate at RegressionMetrics.scala:57	2022/12/08 16:10:23	0.7 s	2/2 (1 skipped)	213/213 (1 skipped)
12	show at MLmodelTrainer.java:57 show at MLmodelTrainer.java:57	2022/12/08 16:10:23	0.2 s	1/1 (1 skipped)	20/20 (1 skipped)
11	show at MLmodelTrainer.java:57 show at MLmodelTrainer.java:57	2022/12/08 16:10:23	91 ms	1/1 (1 skipped)	4/4 (1 skipped)
10	show at MLmodelTrainer.java:57 show at MLmodelTrainer.java:57	2022/12/08 16:10:22	0.8 s	2/2	2/2
9	csv at MLmodelTrainer.java:89 csv at MLmodelTrainer.java:89	2022/12/08 16:10:21	45 ms	1/1	1/1
8	csv at MLmodelTrainer.java:89 csv at MLmodelTrainer.java:89	2022/12/08 16:10:21	35 ms	1/1	1/1
7	count at LinearRegression.scala:953 count at LinearRegression.scala:953	2022/12/08 16:10:21	49 ms	1/1 (2 skipped)	1/1 (201 skipped)
6	count at LinearRegression.scala:953 count at LinearRegression.scala:953	2022/12/08 16:10:21	0.5 s	1/1 (1 skipped)	200/200 (1 skipped)
5	sum at RegressionMetrics.scala:71 sum at RegressionMetrics.scala:71	2022/12/08 16:10:20	0.4 s	1/1 (1 skipped)	200/200 (1 skipped)
4	treeAggregate at RegressionMetrics.scala:57 treeAggregate at RegressionMetrics.scala:57	2022/12/08 16:10:19	0.8 s	2/2 (1 skipped)	213/213 (1 skipped)
3	treeAggregate at WeightedLeastSquares.scala:105 treeAggregate at WeightedLeastSquares.scala:105	2022/12/08 16:10:15	3 s	2/2 (1 skipped)	213/213 (1 skipped)
2	first at LinearRegression.scala:322 first at LinearRegression.scala:322	2022/12/08 16:10:14	1 s	2/2	2/2
1	csv at MLmodelTrainer.java:89 csv at MLmodelTrainer.java:89	2022/12/08 16:10:12	1 s	1/1	1/1
0	csv at MLmodelTrainer.java:89 csv at MLmodelTrainer.java:89	2022/12/08 16:10:08	4 s	1/1	1/1



Learner Lab | EMR – AWS Console | Wine Quality Model - Stages

p-2i8jq6uiynt9m.emrappui-prod.us-east-1.amazonaws.com/shs/history/application_1670512398715_0001/1/stages/

Spark 2.4.8-amzn-2 — Jobs | Stages | Storage | Environment | Executors | SQL — Wine Quality Model application UI

Stages for All Jobs

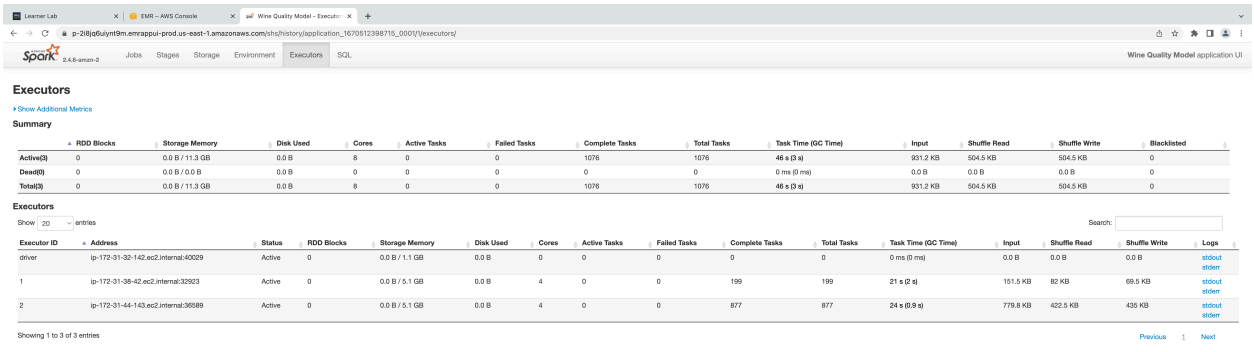
Completed Stages: 23
Skipped Stages: 10

▾ Completed Stages (23)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
32	parquet at LinearRegression.scala:747 +details	2022/12/08 16:10:25	0.6 s	1/1		2026.0 B	186.0 B	
30	parquet at LinearRegression.scala:747 +details	2022/12/08 16:10:24	15 ms	1/1				186.0 B
29	runJob at SparkHadoopWriter.scala:78 +details	2022/12/08 16:10:24	45 ms	1/1		510.0 B		
28	runJob at SparkHadoopWriter.scala:78 +details	2022/12/08 16:10:24	0.1 s	1/1		202.0 B		
27	treeAggregate at RegressionMetrics.scala:57 +details	2022/12/08 16:10:24	84 ms	13/13			79.7 KB	
26	treeAggregate at RegressionMetrics.scala:57 +details	2022/12/08 16:10:23	0.6 s	200/200	8.2 KB		12.2 KB	79.7 KB
24	show at MLmodelTrainer.java:57 +details	2022/12/08 16:10:23	0.2 s	20/20			2.2 KB	
22	show at MLmodelTrainer.java:57 +details	2022/12/08 16:10:23	83 ms	4/4			198.0 B	
20	show at MLmodelTrainer.java:57 +details	2022/12/08 16:10:23	94 ms	1/1			201.0 B	
19	show at MLmodelTrainer.java:57 +details	2022/12/08 16:10:22	0.7 s	1/1	8.3 KB			14.7 KB
18	csv at MLmodelTrainer.java:89 +details	2022/12/08 16:10:21	37 ms	1/1	8.3 KB			
17	csv at MLmodelTrainer.java:89 +details	2022/12/08 16:10:21	32 ms	1/1	8.3 KB			
16	count at LinearRegression.scala:953 +details	2022/12/08 16:10:21	45 ms	1/1			11.5 KB	
13	count at LinearRegression.scala:953 +details	2022/12/08 16:10:21	0.4 s	200/200	217.6 KB			11.5 KB
11	sum at RegressionMetrics.scala:71 +details	2022/12/08 16:10:20	0.4 s	200/200	235.8 KB			
9	treeAggregate at RegressionMetrics.scala:57 +details	2022/12/08 16:10:20	99 ms	13/13			97.2 KB	
8	treeAggregate at RegressionMetrics.scala:57 +details	2022/12/08 16:10:19	0.7 s	200/200	226.1 KB			97.2 KB
6	treeAggregate at WeightedLeastSquares.scala:105 +details	2022/12/08 16:10:19	0.1 s	13/13			207.6 KB	
5	treeAggregate at WeightedLeastSquares.scala:105 +details	2022/12/08 16:10:15	3 s	200/200	1040.0 B		81.5 KB	207.6 KB
3	first at LinearRegression.scala:322 +details	2022/12/08 16:10:15	0.5 s	1/1			380.0 B	
2	first at LinearRegression.scala:322 +details	2022/12/08 16:10:14	0.5 s	1/1	65.8 KB			81.9 KB
1	csv at MLmodelTrainer.java:89 +details	2022/12/08 16:10:12	1 s	1/1	65.8 KB			
0	csv at MLmodelTrainer.java:89 +details	2022/12/08 16:10:08	2 s	1/1	64.0 KB			

▾ Skipped Stages (10)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
31	parquet at LinearRegression.scala:747 +details	Unknown	Unknown	0/1				
25	show at MLmodelTrainer.java:57 +details	Unknown	Unknown	0/1				
23	show at MLmodelTrainer.java:57 +details	Unknown	Unknown	0/1				
21	show at MLmodelTrainer.java:57 +details	Unknown	Unknown	0/1				
15	count at LinearRegression.scala:953 +details	Unknown	Unknown	0/200				
14	first at LinearRegression.scala:322 +details	Unknown	Unknown	0/1				
12	first at LinearRegression.scala:322 +details	Unknown	Unknown	0/1				
10	first at LinearRegression.scala:322 +details	Unknown	Unknown	0/1				
7	first at LinearRegression.scala:322 +details	Unknown	Unknown	0/1				
4	first at LinearRegression.scala:322 +details	Unknown	Unknown	0/1				



Learner Lab | EMR – AWS Console | Wine Quality Model - Executors

p-2i8jq6uiynt9m.emrappui-prod.us-east-1.amazonaws.com/shs/history/application_1670512398715_0001/1/executors/

Spark 2.4.8-amzn-2 — Jobs | Stages | Storage | Environment | Executors | SQL — Wine Quality Model application UI

Executors

▸ Show Additional Metrics

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(3)	0	0.0 B / 11.3 GB	0.0 B	8	0	0	1076	1076	**46 s (3 s)**	931.2 KB	504.5 KB	504.5 KB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(3)	0	0.0 B / 11.3 GB	0.0 B	8	0	0	1076	1076	**46 s (3 s)**	931.2 KB	504.5 KB	504.5 KB	0

Executors

Show 20 entries — Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-172-31-32-142.ec2.internal:40029	Active	0	0.0 B / 1.1 GB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
1	ip-172-31-38-42.ec2.internal:32923	Active	0	0.0 B / 5.1 GB	0.0 B	4	0	0	199	199	**21 s (2 s)**	151.5 KB	82 KB	69.5 KB	stdout stderr
2	ip-172-31-44-143.ec2.internal:36589	Active	0	0.0 B / 5.1 GB	0.0 B	4	0	0	877	877	**24 s (0.9 s)**	779.8 KB	422.5 KB	435 KB	stdout stderr

Showing 1 to 3 of 3 entries — Previous 1 Next

7. Perform wine quality prediction model without Docker on a single EC2 instance.
    - spark-submit --class winequality.WineQualityPrediction WQP-1.0.jar hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/model/

8. Perform wine quality prediction model with Docker on a single EC2 instance.
    - clone GitHub repository using command:
      git clone https://github.com/ms298njit/CS643-PA2-WQP.git
    - Install docker on the EC2 instance
    - Link to docker image using command:
      docker pull ms298/wqp
    - run command to start the spark cluster:
      docker-compose up -d
    - Copy files to HDFS
      docker cp data/model spark-master:/opt/workspace && docker cp data/ValidationDataset.csv spark-master:/opt/workspace/ ValidationDataset.csv
    - Once the cluster is up and running, run command to build ML model and generate output:
      docker run --rm -it --network winequality -v hadoop-distributed-file-system:/opt/workspace --name WQP-test --link spark-master:spark-master ms298/wqp