

## **# CS 643 AWS ML Programming Assignment**

### Github Link:

<https://github.com/John-Santucci/CS-643-Cloud-Computing-Assignment-2>

### Docker Link

[https://hub.docker.com/r/johnsantucci/cs643assignment2/tags?page=1&ordering=last\\_updated](https://hub.docker.com/r/johnsantucci/cs643assignment2/tags?page=1&ordering=last_updated)

## # Completion Steps

### # Docker

Open up Command terminal

Access your file directory

Enter > docker login

Then enter > docker build --tag cs643\_sparkrunner .

Finally enter > docker run --name cs643\_sparkrunner node:latest

## # How to Set up AWS

### # Create EC2 instances:

Under EMR create a cluster with 5 EC2 instances with Amazon Linux AMI.

Under software configuration set the applications to Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.8.2

Under Hardware Configuration set the number of instances to 5.

Under Security and Access, we will proceed without an EC2 key pair.

1 EC2 instance will be the master. The 4 other EC2 instances will be slaves.

For the open ports, select 22 (SSH), 80 (HTTP), and 443 (HTTPS).

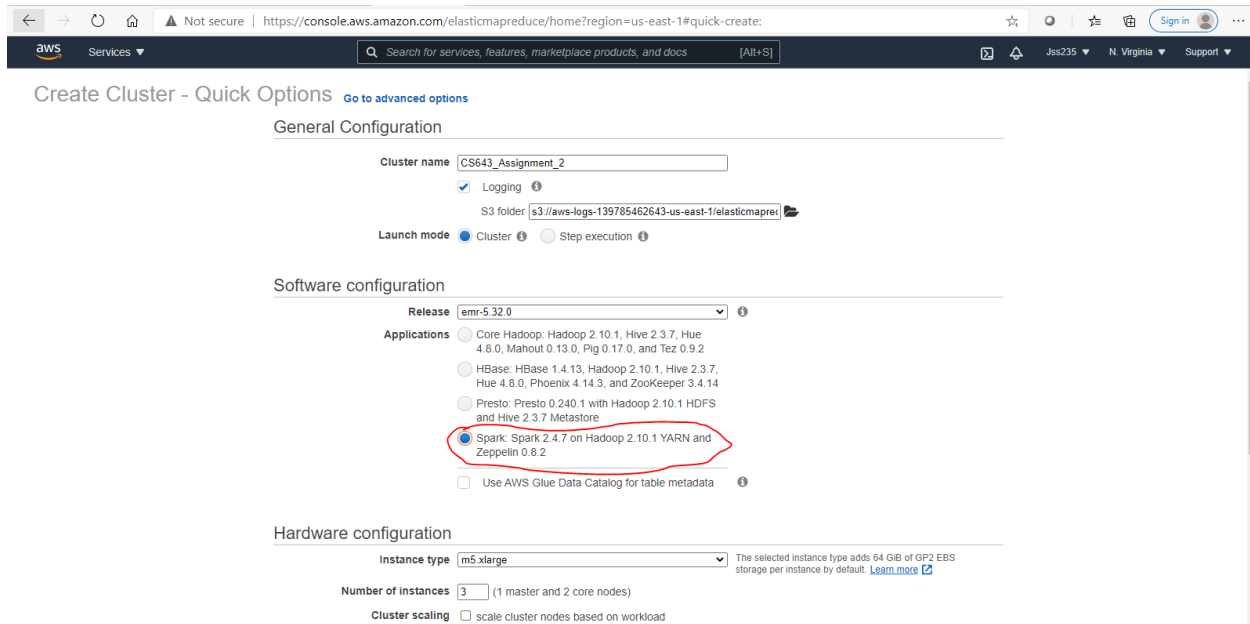
Add SparkRunner.jar and TrainingDataset.csv to S3 buckets

Under the cluster, go to add steps.

Add SparkRunner.jar from the s3 bucket.

SetAction on Failure to Cancel and Wait.

Once the SparkRunner is uploaded it will give the Wine Quality Prediction.





Step type: Custom JAR

Name: SparkRunner

JAR location: s3://jsa235-assignment2/SparkRunner.jar

Arguments:

Action on failure: Cancel and wait

Buttons: Cancel, Add

Concurrency: 1 [Change](#)

After last step completes: Cluster waits

[Add step](#) [Clone step](#) [Cancel step](#)

[View Jobs in t](#)

Filter: <a href="#">All steps</a> <input type="text" value="Filter steps ..."/>		4 steps (all loaded)				
	ID	Name	Status	Start time (UTC-5)	Elapsed time	Log files
<input type="radio"/>	s-1U2EAO6L2EFNA	Custom JAR	Completed	2020-12-21 16:23 (UTC-5)	2 seconds	<a href="#">controller</a>   <a href="#">syslog*</a>   <a href="#">stderr</a>   <a href="#">stdout*</a>
<input type="radio"/>	s-1P8V6K3EEZQA1	Setup hadoop debugging	Completed	2020-12-21 11:41 (UTC-5)	6 seconds	<a href="#">View logs</a>

The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications in Amazon AWS cloud platform. Specifically, you will learn: (1) how to use Apache Spark to train an ML model in parallel on multiple EC2 instances; (2) how to use Spark's MLlib to develop and use an ML model in the cloud; (3) How to use Docker to create a container for your ML model to simplify model deployment.

### # Wine Quality Prediction ML Model

Build a wine quality prediction ML model in Spark over AWS. The model must be trained in parallel using 4 EC2 instances. Then, you need to 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 on Ubuntu Linux. The details of the assignment are presented below:

#### # Input for Model Training:

There are 2 datasets with you for your ML model.

TrainingDataset.csv: you will use this dataset to train the model in parallel on multiple EC2 instances.

ValidationDataset.csv: you will use this dataset to validate the model and optimize its performance (i.e., select the best values for the model parameters).

#### # Input for Prediction Testing:

TestDataset.csv. We will use this file, which has a similar structure with the two datasets above, to test the functionality and performance of your prediction application. Your prediction application should take such a file as input. This file is not shared with you, but you can use the validation dataset to make sure your application works.

### # Model Implementation:

You have to develop a Spark application that uses MLlib to train for wine quality prediction using the training dataset. You will use the validation dataset check the performance of your trained model and to potentially tune your ML model parameters for best performance. You should start with a simple linear regression or logistic regression model from MLlib, but you can try multiple ML models to see which one leads to better performance. For classification models, you can use 10 classes (the wine scores are from 1 to 10). Note: there will be extra-credit for the top 5 applications/students in terms of prediction performance (see below under grading).