# CS 643 AWS ML Programming Assignment

Github Link:
https://github.com/John-Santucci/CS-643-Cloud-Computing-Programming-Assignment-2

Docker Link
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 .

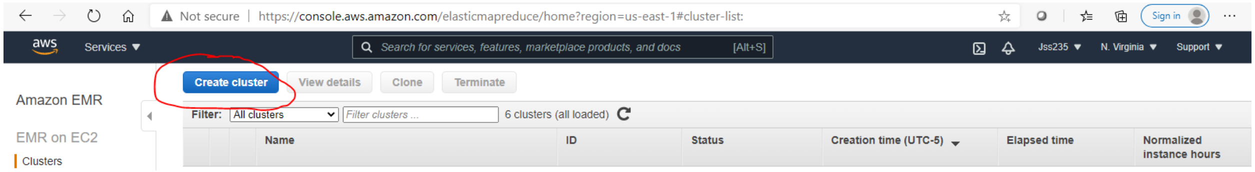
Finally enter > docker run --name cs643_sparkrunner node:latest

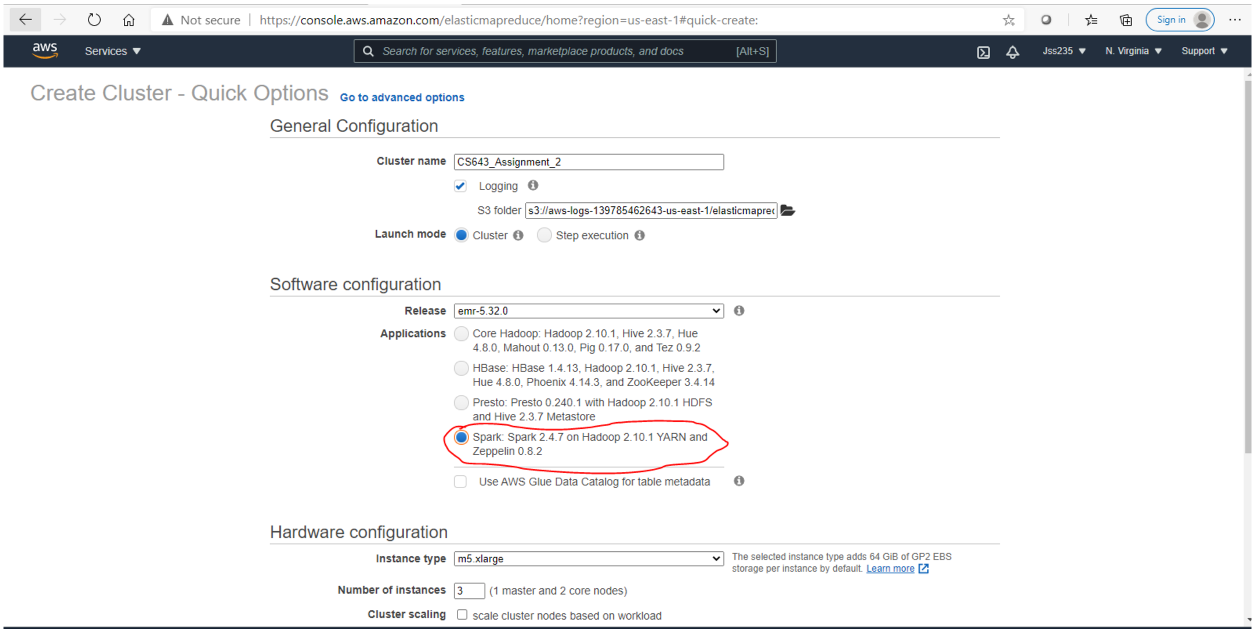
# How to Set up AWS

# 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.




Not secure | https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:
Sign in
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Jss235 ▼
N. Virginia ▼
Support ▼

Create Cluster - Quick Options Go to advanced options

General Configuration

Cluster name: CS643_Assignment_2

☑ Logging

S3 folder: s3://aws-logs-139785462643-us-east-1/elasticmapred

Launch mode: ● Cluster ○ Step execution

Software configuration

Release: emr-5.32.0

Applications:

○ Core Hadoop: Hadoop 2.10.1, Hive 2.3.7, Hue 4.8.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2

○ HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.7, Hue 4.8.0, Phoenix 4.14.3, and ZooKeeper 3.4.14

○ Presto: Presto 0.240.1 with Hadoop 2.10.1 HDFS and Hive 2.3.7 Metastore

● Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.8.2

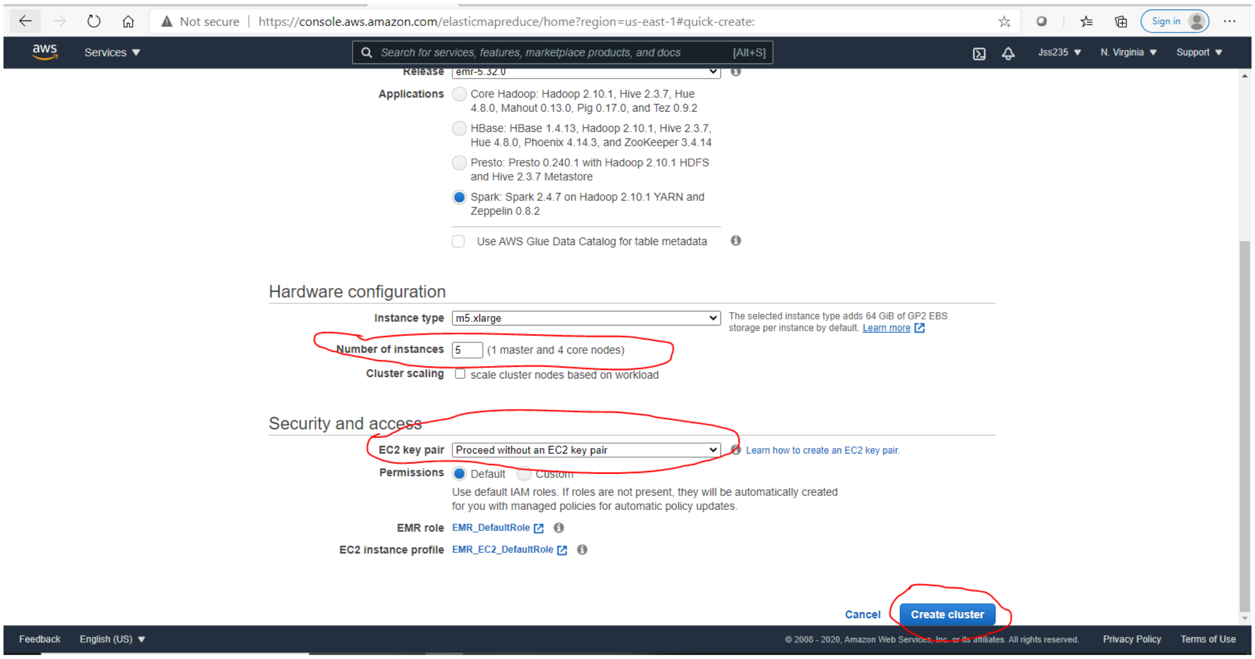
□ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type: m5.xlarge — The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. Learn more

Number of instances: 3 (1 master and 2 core nodes)

Cluster scaling: □ scale cluster nodes based on workload


Not secure | https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:
Sign in
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Jss235 ▼
N. Virginia ▼
Support ▼

Release: emr-5.32.0

Applications:

○ Core Hadoop: Hadoop 2.10.1, Hive 2.3.7, Hue 4.8.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2

○ HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.7, Hue 4.8.0, Phoenix 4.14.3, and ZooKeeper 3.4.14

○ Presto: Presto 0.240.1 with Hadoop 2.10.1 HDFS and Hive 2.3.7 Metastore

● Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.8.2

□ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type: m5.xlarge — The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. Learn more

Number of instances: 5 (1 master and 4 core nodes)

Cluster scaling: □ scale cluster nodes based on workload

Security and access

EC2 key pair: Proceed without an EC2 key pair — Learn how to create an EC2 key pair.

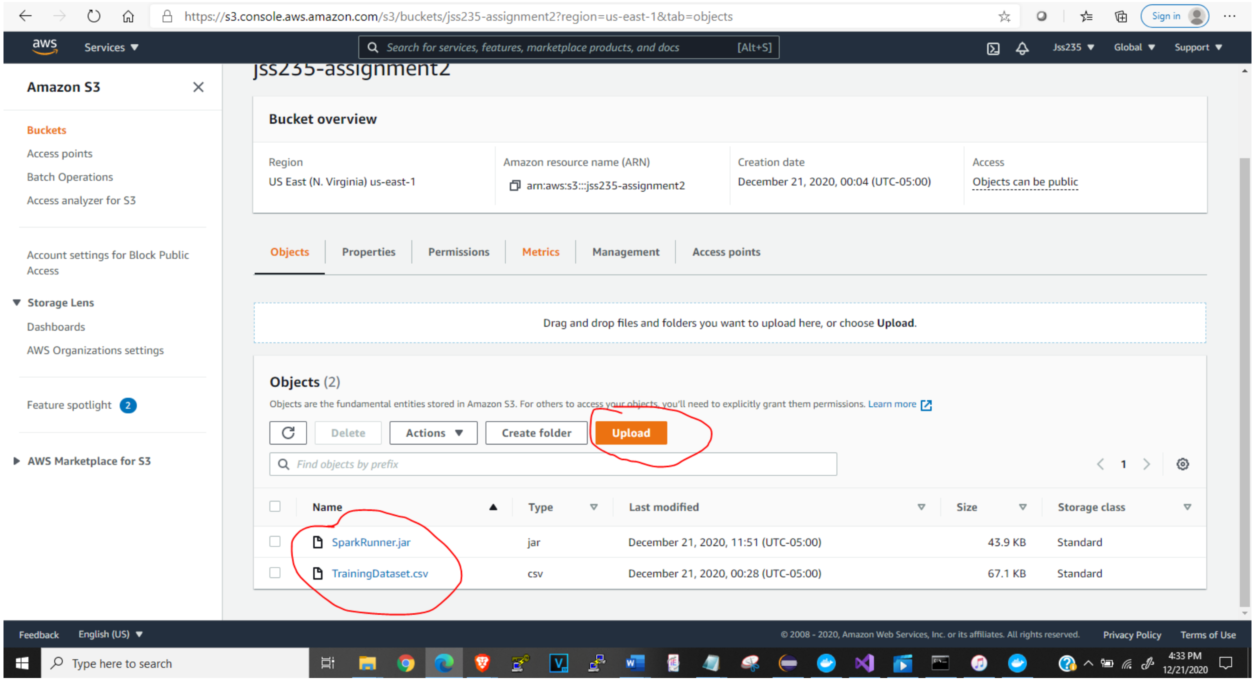
Permissions: ● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole

EC2 instance profile: EMR_EC2_DefaultRole

Cancel
Create cluster
Feedback
English (US) ▼
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



jss235-assignment2

Bucket overview

Region	Amazon resource name (ARN)	Creation date	Access
US East (N. Virginia) us-east-1	arn:aws:s3:::jss235-assignment2	December 21, 2020, 00:04 (UTC-05:00)	Objects can be public

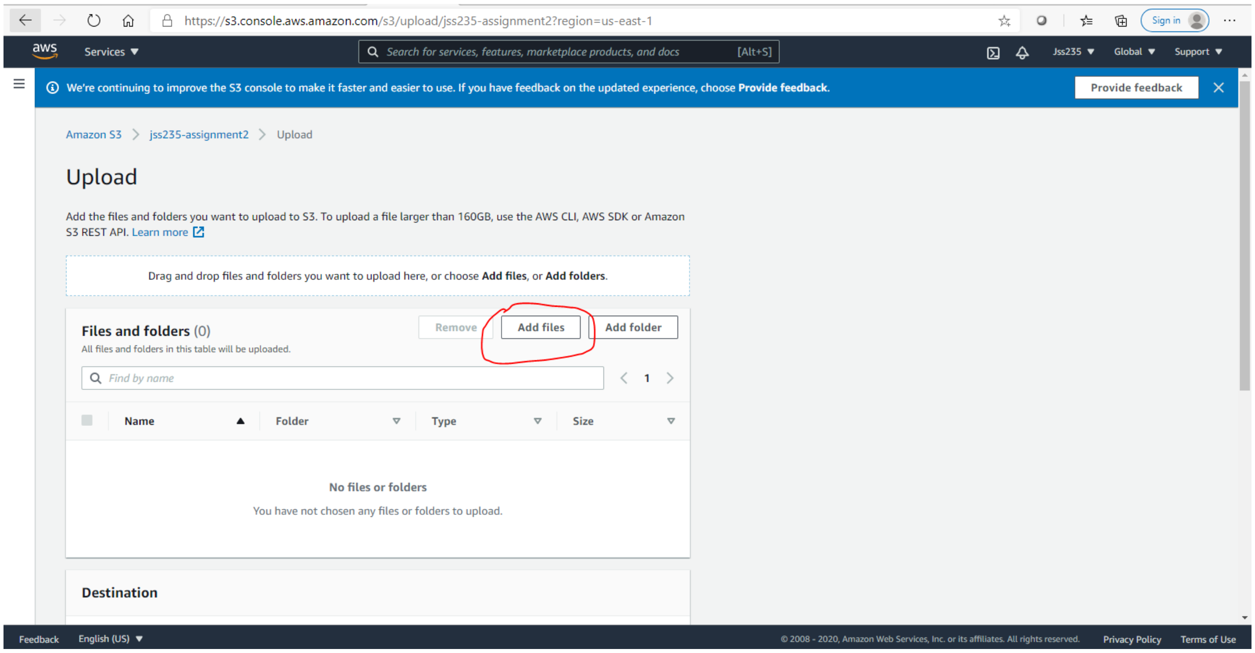
Drag and drop files and folders you want to upload here, or choose **Upload**.

Objects (2)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

☐	Name	Type	Last modified	Size	Storage class
☐	SparkRunner.jar	jar	December 21, 2020, 11:51 (UTC-05:00)	43.9 KB	Standard
☐	TrainingDataset.csv	csv	December 21, 2020, 00:28 (UTC-05:00)	67.1 KB	Standard

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


https://s3.console.aws.amazon.com/s3/upload/jss235-assignment2?region=us-east-1
Sign in
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Jss235 ▼
Global ▼
Support ▼
Provide feedback
Amazon S3 > jss235-assignment2 > Upload
Remove
Add files
Add folder
Find by name
1
Feedback
English (US) ▼
Privacy Policy
Terms of Use

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Upload

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (0)

All files and folders in this table will be uploaded.

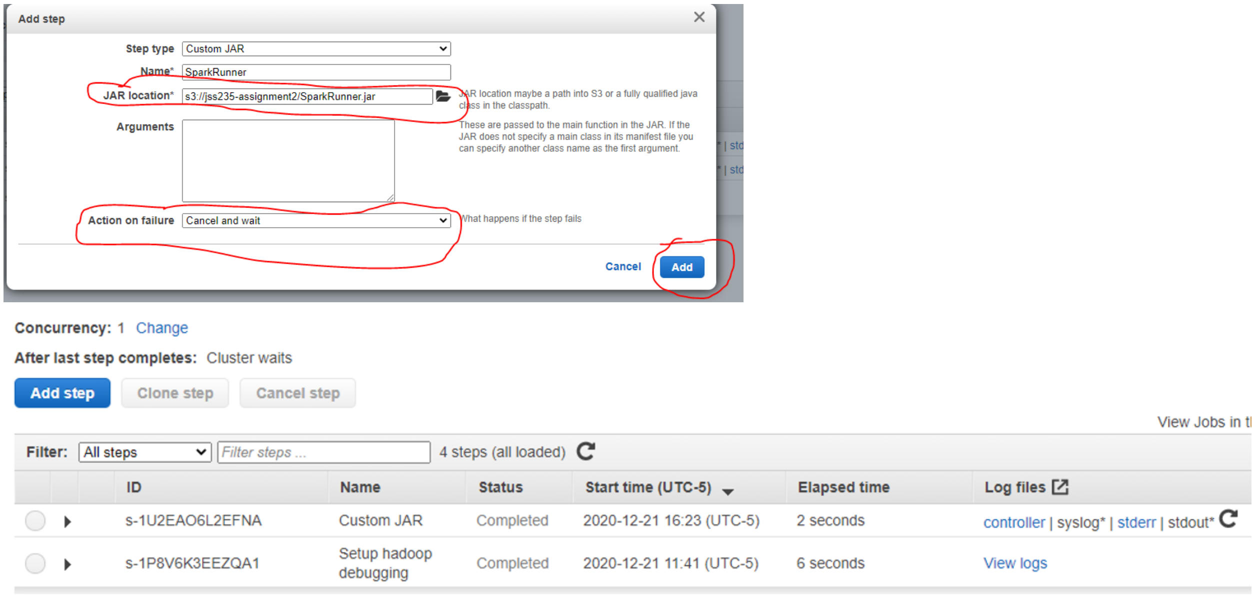
☐	Name	Folder	Type	Size

No files or folders

You have not chosen any files or folders to upload.

Destination

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Add step

Step type: Custom JAR

Name*: SparkRunner

JAR location*: s3://jss235-assignment2/SparkRunner.jar

JAR location maybe a path into S3 or a fully qualified java class in the classpath.

Arguments

These are passed to the main function in the JAR. If the JAR does not specify a main class in its manifest file you can specify another class name as the first argument.

Action on failure: Cancel and wait

What happens if the step fails

Concurrency: 1 Change

After last step completes: Cluster waits

Filter: All steps Filter steps ... 4 steps (all loaded)

ID	Name	Status	Start time (UTC-5)	Elapsed time	Log files
s-1U2EAO6L2EFNA	Custom JAR	Completed	2020-12-21 16:23 (UTC-5)	2 seconds	controller | syslog* | stderr | stdout*
s-1P8V6K3EEZQA1	Setup hadoop debugging	Completed	2020-12-21 11:41 (UTC-5)	6 seconds	View logs

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).