

Programming Assignment-2
CS643-Cloud Computing

Submitted By:
Akash Shrivastava
UCID: AS57

Section-1

GitHub URLs:

Parallel training implementation-

https://github.com/akatast/Wine_quality_prediction/blob/main/wineQPredModelTraining.py

Single machine prediction application-

https://github.com/akatast/Wine_quality_prediction/blob/main/wineQPredModelValidation.py

Docker hub URL:

Docker container for prediction application-

<https://hub.docker.com/repository/docker/as5721/as57dockerpublic>

Command to execute docker container using input file-

```
sudo docker run -it -v `pwd`/TestDataset.csv:/dataset/TestDataset.csv as5721/as57dockerpublic:test-wine-qp /dataset/TestDataset.csv
```

Please make sure docker is started and input file TestDataset.csv is available at the same directory where this command is being submitted.

Section-2

AWS Cloud environment set-up

2.1 AWS educate account:

As very first step, login to AWS educate classroom (vocareum) and click on AWS Console.

2.2 Create EMR cluster:

Step 1: Search for the EMR (Amazon Elastic MapReduce) and click on create cluster:

The screenshot shows the AWS Management Console for Amazon Elastic MapReduce. The left sidebar lists navigation options like 'Amazon EMR', 'EMR Studio', 'EMR on EC2', 'Clusters', 'Notebooks', 'Git repositories', 'Security configurations', 'Block public access', 'VPC subnets', 'Events', 'EMR on EKS', 'Virtual clusters', 'Help', and 'What's new'. The main content area is titled 'Welcome to Amazon Elastic MapReduce' and includes a 'Create cluster' button. Below this, it explains 'How Elastic MapReduce Works' with three steps: Upload (upload data to S3), Create (configure and create cluster), and Monitor (monitor health and progress). The right sidebar provides 'Additional Information' links such as 'More about Elastic MapReduce', 'EMR overview', 'FAQs', 'Pricing', 'More Help Using Elastic MapR', 'Forum', 'Documentation', 'Developer Guide', 'API Reference', 'EMR on GitHub', and 'Help portal'.

Step 2: Give the cluster name and select Spark as application:

The screenshot shows the 'Create Cluster - Quick Options' page in the AWS console. The 'General Configuration' section includes a 'Cluster name' field with the value 'CS643_Akash', a checked 'Logging' checkbox, and an 'S3 folder' field with the value 's3://aws-logs-904705107994-us-east-1/elasticmaprec'. The 'Launch mode' is set to 'Cluster'. The 'Software configuration' section shows the 'Release' as 'emr-5.33.0' and 'Applications' as 'Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.9.0'. There are also links to 'Go to advanced options' and 'Use AWS Glue Data Catalog for table metadata'.

As mentioned in the requirement, use number of instances as 4 (one master and four core) and select the key pair:

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 CCAkash2021 ▼ [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](#) ℹ

EC2 instance profile [EMR_EC2_DefaultRole](#) ℹ

Step 3: Then click on create cluster and you'll get following screen with "Starting" message:

The screenshot shows the AWS Management Console interface for an Amazon EMR cluster. The cluster name is 'CS643_Akash' and its status is 'Starting'. The console displays various tabs for cluster management, including Summary, Application user interfaces, Monitoring, Hardware, Configurations, Events, Steps, and Bootstrap actions. The Summary tab is active, showing details such as the cluster ID (j-38T72M69328PR), creation date (2021-07-24 17:44 UTC-4), and the number of instances (1 master and 4 core nodes). The Configuration details section shows the release label (emr-5.33.0), Hadoop distribution (Amazon), and applications (Spark 2.4.7, Zeppelin 0.9.0). The Network and hardware section shows the availability zone (us-east-1a) and the subnet (subnet-c5a9e3e4). The Security and access section shows the key name (CCAkash2021) and the EC2 instance profile (EMR_EC2_DefaultRole).

Cluster: CS643_Akash Starting

Summary

- ID: j-38T72M69328PR
- Creation date: 2021-07-24 17:44 (UTC-4)
- Elapsed time: 0 seconds
- After last step completes: Cluster waits
- Termination protection: Off [Change](#)
- Tags: -- [View All / Edit](#)
- Master public DNS: --

Configuration details

- Release label: emr-5.33.0
- Hadoop distribution: Amazon
- Applications: Spark 2.4.7, Zeppelin 0.9.0
- Log URI: s3://aws-logs-904705107994-us-east-1/elasticmapreduce/
- EMRFS consistent view: Disabled
- Custom AMI ID: --

Application user interfaces

- Persistent user interfaces [🔗](#): --
- On-cluster user -- [🔗](#) interfaces [🔗](#):

Network and hardware

- Availability zone: --
- Subnet ID: [subnet-c5a9e3e4](#) [🔗](#)
- Master: **Provisioning** 1 m5.xlarge
- Core: **Provisioning** 4 m5.xlarge
- Task: --
- Cluster scaling: Not enabled

Security and access

- Key name: CCAkash2021
- EC2 instance profile: EMR_EC2_DefaultRole
- EMR role: EMR_DefaultRole
- Visible to all users: All [Change](#)

Step 4: Go to security group of master and click on inbound rules:

Search for services, features, marketplace products, and docs [Alt+S] vocstartsoft/user987826=as57@njit.edu @ 9047-0510-7994 N. Virginia

Security Groups (1/2) Info

Filter security groups

search: sg-00c3995d52266b936 Clear filters

	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input checked="" type="checkbox"/>	-	sg-00c3995d52266b936	ElasticMapReduce-mas...	vpc-a17ee1dc	Master group for Elasti...	904705107994
<input type="checkbox"/>	-	sg-0ff029d2f73047ac3	ElasticMapReduce-slave	vpc-a17ee1dc	Slave group for Elastic ...	904705107994

sg-00c3995d52266b936 - ElasticMapReduce-master

Details **Inbound rules** Outbound rules Tags

Inbound rules (18)

Filter security group rules

Manage tags Edit inbound rules

Click on 'Edit inbound rules' and add a new rule with type = SSH, port range=22 and source = anywhere

sgr-09e41bf98c402a4d7 Custom TCP TCP 8443 Custom 72.21.217.0/24 Delete

- SSH TCP 22 Anywh... 0.0.0.0/0 Delete

Add rule

Cancel Preview changes Save rules

Step 5: AWS gives the steps to Connect to the Master Node Using SSH. Use the following steps to connect to the master node:


SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. [Learn more](#).

Windows

Mac / Linux

1. Download PuTTY.exe to your computer from:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> 
2. Start PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-44-195-60-199.compute-1.amazonaws.com**
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**CCAakash2021.ppk**) used to launch the cluster.
7. Click Open.
8. Click Yes to dismiss the security alert.

Step 6: After successful connection we'll get following screen:

[illegible]

Section-3

S3 bucket and file storage

After termination of EMR cluster we lose all the data so we can use S3 bucket to store the files beyond one session/longer-term.

3.1 Create a new S3 bucket:

Create a S3 bucket to store the files-

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a user profile. Below the navigation bar is a blue banner with a message: '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](#).' The main content area is titled 'General configuration'. It contains a 'Bucket name' field with the value 'akash.cs643', a note that the name must be unique and not contain spaces or uppercase letters, and a link to 'See rules for bucket naming'. Below this is the 'AWS Region' dropdown menu, currently set to 'US East (N. Virginia) us-east-1'. At the bottom, there's a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button.

We'll get following success message-

The screenshot shows the AWS S3 console after successful bucket creation. A green banner at the top displays a success message: 'Successfully created bucket "akash.cs643"'. Below this is a blue banner with a tip: 'How to optimize your costs on S3.' The main content area is titled 'Account snapshot' and includes a 'View Storage Lens dashboard' button. It shows metrics for 'Total storage' (722.7 KB), 'Object count' (129), and 'Avg. object size' (5.6 KB). Below this is a section for 'Buckets (2)' with a search bar and a table of buckets. The table has columns for Name, AWS Region, Access, and Creation date. The first bucket listed is 'akash.cs643' in the 'US East (N. Virginia) us-east-1' region, with 'Bucket and objects not public' access and a creation date of 'July 24, 2021, 16:12:53 (UTC-04:00)'.

Name	AWS Region	Access	Creation date
akash.cs643	US East (N. Virginia) us-east-1	Bucket and objects not public	July 24, 2021, 16:12:53 (UTC-04:00)

3.2 Upload the files:

Upload the training and validation datasets to S3 bucket-

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a user profile. Below the navigation bar, a blue banner indicates a feedback prompt. A green banner below that states 'Upload succeeded' with a link to 'View details below'. The main content area is divided into two tabs: 'Files and folders' (selected) and 'Configuration'. Under the 'Files and folders' tab, a summary section shows the destination 's3://akash.cs643', 'Succeeded' status with '2 files, 75.7 KB (100.00%)', and 'Failed' status with '0 files, 0 B (0%)'. Below this, a table lists the uploaded files:

Name	Folder	Type	Size	Status
TrainingDataset.csv	-	application/vnd.ms-excel	67.2 KB	✓ Succeeded
ValidationDataset.csv	-	application/vnd.ms-excel	8.6 KB	✓ Succeeded

Section-4

Model training and validation

4.1 Parallel training implementation (Train the model):

Connect to the Master node, move the model training program to Master node and give the execute permission.

Please check page-1 for GitHub link of Parallel training implementation.

Execute the training program using following command-

```
Spark-submit wineQPredModelValidation.py
```

I am printing training error and F1 score after completion of training.

After successful execution, check the S3 bucket if model has been stored-

Objects (3)

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

Copy S3 URI Copy URL Download Open Delete Actions Create f

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	TrainingDataset.csv	csv	July 25, 2021, 22:43:45 (UTC-04:00)
<input type="checkbox"/>	ValidationDataset.csv	csv	July 25, 2021, 22:43:45 (UTC-04:00)
<input type="checkbox"/>	wineQpredmodel.model/	Folder	-

4.2 Single machine prediction application:

Please check page-1 for GitHub link of Single machine prediction application.

Execute the validation program:

```
spark-submit wineQPredModelValidation.py
```

Section-5

Create Docker Container

Connect to EC-2 instance of master and perform following steps:

Install spark-

```
pip install --user pyspark
```

Set the path-

```
export PYSPARK_PYTHON=/usr/bin/python3
export PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```

Install and start the Docker-

```
sudo yum update -y
sudo amazon-linux-extras install docker
sudo yum install docker
sudo service docker start
sudo usermod -a -G docker Hadoop
```

Change the permission-

```
sudo chmod 666 /var/run/docker.sock
```


Build the image (before executing this command, make sure the trained model, model testing program and the dockerfile are present on same directory)-

```
sudo docker build -t imagename .
```

List the available images and verify-

```
sudo docker image ls
```

Login to docker hub using following command, username, and password-

```
docker login
```

Go to dockerhub website and create a new public repo.

On the EC2 terminal tag the created docker image-

```
docker tag imagename:version username/repo:tagname
```

From EC2 push the image to docker hub repo-

```
docker push username/repo:tagname
```

Now docker container is ready, anyone can issue a pull request and execute it in same OS (Linux) using the input file TestDataset.csv

Docker container for prediction application

Following is command to execute the container-

```
sudo docker run -it -v `pwd`/TestDataset.csv:/dataset/TestDataset.csv  
as5721/as57dockerpublic:test-wine-qp /dataset/TestDataset.csv
```