

CS643861- Programming Assignment-2

Name: Lakshman Palli

UCID: vp692

Wine Quality Prediction AWS Spark Application:

Pa2Winepred: This project requires the creation of a Python application that uses the PySpark interface.

The application is running on an Amazon Web Services (AWS) Elastic MapReduce (EMR) cluster. The primary goal is to simultaneously train a machine learning model on EC2 instances to predict wine quality using publicly available data. The trained model is then used to predict the wine's quality. Docker is used to create a container image for the trained machine learning model, which simplifies the deployment process.

Link for GitHub:

<https://github.com/LakshmanPalli/wine-quality-prediction>

Link for Docker:

<https://hub.docker.com/layers/lakshmanpalli692/qulwinepred/latest/images/sha256-05319e880738d975a2bdf790c22eaacbb67d9add7b189c47511641d834d4f23c?context=explore>

Steps for the Execution of Wine Quality Prediction AWS Spark Application:

1. Create a Key-pair for the EMR Cluster: Go to EC2/Network/Key-pairs

Use the format of .pem and download the keypair

Created key pair as CS643key692.pem

2. Create an S3 bucket

We must create an S3 bucket in aws: cs643winequlpred2

3. Next, create an EMR cluster through the EMR console.

4. To create the spark in the AWS instance, use the EMR console:

Create the spark cluster by using the EMR console, and create the 4 instances:

Name and application:

Name: Wine_qp

Amazon EMR release: EMR-5.33.0

Application bundle: Hadoop 2.10.1, Spark 2.4.7, Zippeline 0.9.0, and Yarn

CS643861- Programming Assignment-2

Clone "Wine_qp" [Info](#)

▼ Name and applications - required [Info](#)
Name your cluster and choose the applications that you want to install to your cluster.

Name
Wine_qp_clone

Amazon EMR release [Info](#)
A release contains a set of applications which can be installed on your cluster.
emr-5.33.0

Application bundle

Spark	Core Hadoop	HBase	Presto	Custom

☐ Flink 1.12.1
☐ HCatalog 2.3.7
☐ Hue 4.9.0
☐ Livy 0.7.0
☒ Oozie 5.2.0
☐ Presto 0.245.1
☒ TensorFlow 2.4.1
☒ ZooKeeper 3.4.14

☐ Ganglia 3.7.2
☒ Hadoop 2.10.1
☐ JupyterEnterpriseGateway 2.1.0
☐ MXNet 1.7.0
☐ Phoenix 4.14.3
☒ Spark 2.4.7
☐ Tez 0.9.2

☐ HBase 1.4.13
☐ Hive 2.3.7
☐ JupyterHub 1.2.2
☐ Mahout 0.13.0
☐ Pig 0.17.0
☐ Sqoop 1.4.7
☒ Zeppelin 0.9.0

AWS Glue Data Catalog settings
Use the AWS Glue Data Catalog to provide an external metastore for your application.
☐ Use for Spark table metadata

Custom Amazon Machine Image (AMI) [Info](#)
Choose or enter an AMI ID

☒ Update all installed packages on reboot

Note: It says Clone "wine_qp_clone" because I cloned the previous configuration instead of starting from scratch to save time.

Cluster Configuration:

CS643861- Programming Assignment-2

aws

Search [Alt+S]

N. Virg

voclabs/user3054027=Lakshman_Palli @ 7140

Amazon EMR

EMR on EC2: Clusters

Create cluster

▼ Cluster configuration - **required** Info

Choose a configuration method for the primary, core, and task node groups for your cluster.

☒ **Uniform instance groups**
Choose the same EC2 instance type and purchasing option (On-Demand or Spot) for all nodes in your node group. [Learn more](#)

☐ **Flexible instance fleets**
Choose from the widest variety of provisioning options for the EC2 instances in your cluster. Diversify instance types and purchasing options, and use an allocation strategy. [Learn more](#)

Uniform instance groups

Primary

Choose EC2 instance type

m1.medium
1 vCore 3.75 GiB memory
410 GiB storage On-Demand price: -
Lowest Spot price: -

Actions ▼

☐ Use high availability
Launch highly available, more resilient cluster with three primary nodes on On-Demand Instances. This configuration applies for the lifetime of your cluster. [Learn more](#)

► **Node configuration - optional**

Core

Choose EC2 instance type

m1.medium
1 vCore 3.75 GiB memory
410 GiB storage On-Demand price: -
Lowest Spot price: -

Actions ▼

► **Node configuration - optional**

Task 1 of 1

Name

Task - 1

Choose EC2 instance type

Remove instance group

CloudShell

Feedback

Privacy

Terms

Cookie preferences

Cluster Scaling and Provisioning:

CS643861- Programming Assignment-2

aws

Search [Alt+S]

N. Virg

voclabs/user3054027=Lakshman_Palli @ 7140

Amazon EMR

EMR on EC2: Clusters

Create cluster

▼ Cluster scaling and provisioning - required Info

Choose how Amazon EMR should size your cluster.

Choose an option

☒ Set cluster size manually
Use this option if you know your workload patterns in advance.

☐ Use EMR-managed scaling
Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.

☐ Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

Provisioning configuration

Set the size of your core and task instance groups. Amazon EMR attempts to provision this capacity when you launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Task - 1	m1.medium	3	<input type="checkbox"/>
Core	m1.medium	1	<input type="checkbox"/>

▼ Networking - required Info

Choose the network settings that determine how you and other entities communicate with your cluster.

Virtual private cloud (VPC) Info

vpc-0e6dc5513be721a79

Browse

Create VPC

Subnet Info

subnet-009f863b84530a304

Browse

Create subnet

► EC2 security groups (firewall)

► Steps (0) Info

Use commands and scripts to tell your cluster where to find and how to process your data. Steps run consecutively unless you enable the Concurrency option.

Remove

Edit

Add

CloudShell

Feedback

Privacy

Terms

Cookie preferences

Networking & Cluster Termination:

CS643861- Programming Assignment-2

aws

Search [Alt+S]

N. Virg

voclabs/user3054027=Lakshman_Palli @ 7140

Amazon EMR

EMR on EC2: Clusters

Create cluster

▼ Networking - *required* Info

Choose the network settings that determine how you and other entities communicate with your cluster.

Virtual private cloud (VPC) Info

vpc-0e6dc5513be721a79

Browse

Create VPC

Subnet Info

subnet-009f863b84530a304

Browse

Create subnet

► EC2 security groups (firewall)

► Steps (0) Info

Remove Edit Add

Use commands and scripts to tell your cluster where to find and how to process your data. Steps run consecutively unless you enable the Concurrency option.

▼ Cluster termination and node replacement Info

Choose termination settings and protect your cluster from accidental shutdown.

Termination option

☒ Manually terminate cluster

☐ Automatically terminate cluster after last step ends

☐ Automatically terminate cluster after idle time (Recommended)

☒ Use termination protection

Protects your cluster from accidental termination. If on, you must first turn off protection to terminate the cluster. We recommend turning on termination protection for your long running clusters.

Unhealthy node replacement - new Info

☐ Turn on

Amazon EMR gracefully stops processes on unhealthy nodes to minimize data loss and job interruptions. It quickly replaces unhealthy nodes with new EC2 instances to keep your jobs running smoothly.

☒ Turn off

Amazon EMR adds unhealthy nodes to a denylist while keeping them in the cluster, allowing you continued access for troubleshooting.

► Bootstrap actions (0) Info

Remove Edit Add

Security Configuration and EC2 Key pair & Identity and access management(IAM) roles:

CS643861- Programming Assignment-2

Security configuration and EC2 key pair [Info](#)
Choose a security configuration or create a new one that you can reuse with other clusters.

Security configuration
Select your cluster encryption, authentication, and instance metadata service settings.

[Refresh](#) [Browse](#) [Create security configuration](#)

Amazon EC2 key pair for SSH to the cluster [Info](#)
 [X](#) [Browse](#) [Create key pair](#)

Identity and Access Management (IAM) roles - required [Info](#)
Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role [Info](#)
The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

☒ **Choose an existing service role**
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

☐ **Create a service role**
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role
 [Refresh](#)

EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

☒ **Choose an existing instance profile**
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

☐ **Create an instance profile**
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

Instance profile
 [Refresh](#)

[CloudShell](#) [Feedback](#) [Privacy](#) [Terms](#) [Cookie preferences](#)

We can follow the above steps to create EMR cluster for the instances

CS643861- Programming Assignment-2

The screenshot displays the AWS Management Console interface for an Amazon EMR cluster. At the top, a green banner indicates that the cluster "Wine_qp_clone" has been successfully created. Below this, the cluster name "Wine_qp_clone" is shown along with its update status and action buttons: "Terminate", "Clone in AWS CLI", and "Clone".

The main content area is divided into several sections:

- Summary**: A table-like view providing key details about the cluster.

Cluster info	Applications	Cluster management	Status and time
Cluster ID j-7WWZCSFX2W6O	Amazon EMR version emr-5.33.0	Log destination in Amazon S3 aws-logs-714080349538-us-east-1/elasticmapreduce	Status Starting
Cluster configuration Instance groups	Installed applications Hadoop 2.10.1, Oozie 5.2.0, Spark 2.4.7, TensorFlow 2.4.1, Zeppelin 0.9.0, ZooKeeper 3.4.14	Primary node public DNS -	Creation time December 01, 2024, 00:05 (UTC-05:00)
Capacity 1 Primary 1 Core 3 Task			Elapsed time -19 seconds
- Properties**: A tab showing cluster logs and S3 location.
 - Cluster logs**: Archive log files to Amazon S3 (Turned on). Amazon S3 location: [s3://aws-logs-714080349538-us-east-1/elasticmapreduce/](#). Encryption for logs: Turned off.
- Cluster termination and node replacement**: A tab showing termination options and settings.
 - Termination option**: Manually terminate cluster.
 - Idle time**: -
 - Termination protection**: On.
 - Unhealthy node replacement**: Off.
- Network and security**: A tab for network and security settings.

5. Now we are training the ML model into spark cluster with ec2 instances in parallel locally without docker:

1. Now the cluster will accept the tasks to run the ML model

Need to connect the Master instance in the Terminal:

```
ssh -i "CS643KEY692.pem" ec2-52-200-8-59.compute-1.amazonaws.com
```

and it is successfully logged in.

2. After the login of the Master instance then change the root by using

```
Sudo su
```

CS643861- Programming Assignment-2

[illegible]

3. Submit the task by the command:

```
spark-submit s3://cs643winequlpred2/winequilityprediction.py
```

4. The trace status for the above tasks is then displayed. If the status is a success, a test .model is in the S3 bucket. s3://cs643winequlpred2

6. Now we are running ML model using the Docker:

1. Create a docker account and sign up.
2. After the successful login then download and set up the docker in your local system
3. Install the docker
4. Login to the docker in the power shell by the command

docker login

Pwd

5. After login you need to build the image:

```
docker build -t winequlpred .
```


CS643861- Programming Assignment-2

```
Administrator: Windows PowerShell
PS C:\winpredaul> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\winpredaul> docker build -t predquility .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
              Install the buildx component to build images with BuildKit:
              https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 13.23MB
Step 1/26 : FROM centos:7
--> eab6ee3f4d9d
Step 2/26 : RUN yum -y update && yum -y install python3 python3-dev python3-pip python3-virtualenv java-1.8.0-openjdk wget
--> f5d251b26687
--> Using cache
Step 3/26 : RUN python -V
--> Using cache
--> ae540e6a1b0f
Step 4/26 : RUN python3 -V
--> Using cache
--> 37d2ba512a92
Step 5/26 : ENV PYSPARK_DRIVER_PYTHON python3
--> Using cache
--> 87082aeb8dc9
Step 6/26 : ENV PYSPARK_PYTHON python3
--> Using cache
--> 3cc27a382143
Step 7/26 : RUN pip3 install --upgrade pip
--> Using cache
--> 4ef02a297054
Step 8/26 : RUN pip3 install numpy panda
--> Using cache
--> 051630ba3664
Step 9/26 : RUN pip3 install pandas
--> Using cache
--> 99c207b7e4e3
Step 10/26 : RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz" && mkdir -p /opt/spark && tar -xf apache-spark.tgz -C /opt/spark --strip-c
omponents=1 && rm apache-spark.tgz
--> Using cache
--> 1b0e4863d33e
Step 11/26 : RUN ln -s /opt/spark-3.1.2-bin-hadoop2.7 /opt/spark
--> Using cache
--> h620b56b768
Step 12/26 : RUN (echo 'export SPARK_HOME=/opt/spark' >> ~/.bashrc && echo 'export PATH=$SPARK_HOME/bin:$PATH' >> ~/.bashrc && echo 'export PYSPARK_PYTHON=python3' >> ~/.bashrc)
--> Using cache
--> 857956561861
Step 13/26 : RUN mkdir /code
--> Using cache
--> bc12b784f139
Step 14/26 : RUN mkdir /code/data
```

6. The push and pull into the docker hub repository:

PUSH:

```
docker tag qulwinepred lakshmanpalli692/qulwinepred
```

```
docker push lakshmanpalli692/qulwinepred
```

PULL:

```
docker pull lakshmanpalli692/qulwinepred
```

7. Place your test data file in a designated folder known as "dir." Mount this directory with the Docker container, then run the container with the following command.

```
docker run -v C:\Pa2\data\csv winequlpred testdata.csv
```

CS643861- Programming Assignment-2

```
---Input file for test data is---
data/csv/testdata.csv
-----
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|density| pH|sulphates|alcohol|quality|
|features|label|rank|Prediction|probability|prediction|
-----
| 8.9| 0.22| 0.48| 1.8| 0.077| 29.0| 60.0| 0.9968| 3.39| 0.53| 9.4|
6.0|[8.9,0.22,0.48,1....]| 1.0|[3.48851627289842...|[0.06977628545796...]| 1.0|
7.6| 0.39| 0.31| 2.2| 0.002| 23.0| 71.0| 0.9982| 3.52| 0.65| 9.7|
5.0|[7.6,0.39,0.31,2....]| 0.0|[48.1243835979459...|[0.96248767015891...]| 0.0|
7.9| 0.43| 0.21| 1.6| 0.186| 18.0| 37.0| 0.9966| 3.17| 0.91| 9.5|
5.0|[7.9,0.43,0.21,1....]| 0.0|[48.1539062576703...|[0.96387888515340...]| 0.0|
8.5| 0.49| 0.11| 2.2| 0.004| 9.0| 67.0| 0.9968| 3.17| 0.53| 9.4|
5.0|[8.5,0.49,0.11,2....]| 0.0|[47.6785761357896...|[0.95357152271419...]| 0.0|
6.9| 0.4| 0.14| 2.4| 0.085| 21.0| 40.0| 0.9968| 3.43| 0.63| 9.7|
6.0|[6.9,0.4,0.14,2.4...]| 1.0|[1.82872254349815...|[0.03657445086980...]| 1.0|
-----
only showing top 5 rows
None
Test Accuracy of wine prediction model = 0.983588922595778
Weighted f1 score of wine prediction model = 0.9776578895168527
PS C:\Pa2Winepred>
```

Conclusion: As shown in the above image, got an accuracy of ~98% while predicting the wine quality.