

# Wine Quality Prediction with Spark on AWS

---

## Overview

This project implements a wine quality prediction ML model using Apache Spark on AWS EMR. The model is trained in parallel on 4 EC2 instances and deployed for prediction on a single instance, both with and without Docker.

---

## Assignment Requirements

- **Training:** Train a model on TrainingDataset.csv using 4 EC2 instances.
  - **Validation:** Evaluate and tune the model using ValidationDataset.csv.
  - **Prediction:** Perform predictions on a single EC2 instance, outputting the F1 score
    - Model used Logistic Regression.
  - **Docker:** Containerize the prediction app for easy deployment.
  - **Implementation:** Java, Ubuntu Linux, Spark MLlib.
- 

## Important Link:

- **GitHub Repo:** [click here](#).
  - **Docker Hub:** [click here](#).
- 

## Project Structure

- WineTrainApp.java: Training application code (Logistic Regression).
  - WinePredictApp.java: Prediction application code.
  - pom.xml: Maven configuration.
  - Dockerfile: Docker setup for prediction app.
  - README.md: Instruction to setup the project
- 

## Technologies Used:

- **Apache Spark:** For distributed data processing and machine learning.
  - **Amazon AWS (EMR, S3):** To train models in a distributed cluster environment.
  - **Docker:** To package the prediction application into a container.
-

# Development Setup:

---

## Desktop Setup - Windows :

### Step 1 : Install or update Java

- Install oracle jdk 8 or openjdk 8 and configure the JAVA\_HOME environment variable.
- update PATH variable to %JAVA\_HOME%\bin; JAVA\_HOME: PATH:
- open the command prompt and check the java version to verify the installation.

```
Microsoft Windows [Version 10.0.26100.3915]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vishal>java -version
java version "21" 2023-09-19 LTS
Java(TM) SE Runtime Environment (build 21+35-LTS-2513)
Java HotSpot(TM) 64-Bit Server VM (build 21+35-LTS-2513, mixed mode, sharing)

C:\Users\Vishal>
```

### Step 2 : Install Maven

- Get it from official site: <https://maven.apache.org/download.cgi>
- Download the binary zip archive (e.g., apache-maven-3.9.6-bin.zip).
- Extract the ZIP file to a location, e.g., C:\apache-maven-3.9.6
- Under System Variables:
  - Click New:
    - Variable name: MAVEN\_HOME
    - Variable value: C:\apache-maven-3.9.6
- Update the Path variable:
  - Edit the Path variable in System Variables and Add "C:\apache-maven-3.9.6\bin"
- Verify installation of Maven using mvn --version command.

```
C:\Users\Vishal>mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\Program Files\Apache\Maven\apache-maven-3.9.9-bin\apache-maven-3.9.9
Java version: 21, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_US, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Vishal>
```

### Step 3 : Create Project Using Maven

#### Commands:

- mkdir CC-CS643-Prgm-Assgn-2
- cd CC-CS643-Prgm-Assgn-2
- mvn archetype:generate -DgroupId=com.wine -DartifactId=wine-ml-spark -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
- cd wine-ml-spark

### Step 4 : Crated the following java files in src\main\java\com\wine\

- WinePrediction.java – To train and save model

- **WinePredictApp.java – To evaluate saved model**

*Lot of additional files were created/updated like POM file, Dockerfile, etc everything is available on GitHub Repo.*

## Step 5 : Build the Project

- mvn clean package : will Output: JAR in target/wine-ml-spark-1.0-SNAPSHOT.jar

## Step 6 : Push everything on github

## Cloud Setup :

### 1 : Create an EMR Cluster :

#### Step 1 : Enter basic Information:

- Give cluster Name
- Select the Amazon EMR Release as emr—6.15.0 (stable version)
- Check services like spark, Hadoop, Hive (all other are optional)

#### Snippet:

**Name and applications - required** Info  
Name your cluster and choose the applications that you want to install on your cluster.

**Name**  
wine-ml-emr-cc

**Amazon EMR release** Info  
A release contains a set of applications which can be installed on your cluster.  
emr-6.15.0

**Application bundle**

Spark Interactive	Core Hadoop	Flink	HBase	Presto	Trino	Custom
-------------------	-------------	-------	-------	--------	-------	--------

**Applications**

<input type="checkbox"/> Flink 1.17.1 <input type="checkbox"/> HCatalog 3.1.3 <input type="checkbox"/> Hue 4.11.0 <input checked="" type="checkbox"/> Livy 0.7.1 <input type="checkbox"/> Phoenix 5.1.3 <input checked="" type="checkbox"/> Spark 3.4.1 <input type="checkbox"/> Tez 0.10.2 <input type="checkbox"/> ZooKeeper 3.5.10	<input type="checkbox"/> Ganglia 3.7.2 <input checked="" type="checkbox"/> Hadoop 3.3.6 <input checked="" type="checkbox"/> JupyterEnterpriseGateway 2.6.0 <input type="checkbox"/> MXNet 1.9.1 <input type="checkbox"/> Pig 0.17.0 <input type="checkbox"/> Sqoop 1.4.7 <input type="checkbox"/> Trino 426	<input type="checkbox"/> HBase 2.4.17 <input checked="" type="checkbox"/> Hive 3.1.3 <input type="checkbox"/> JupyterHub 1.5.0 <input type="checkbox"/> Oozie 5.2.1 <input type="checkbox"/> Presto 0.283 <input type="checkbox"/> TensorFlow 2.11.0 <input type="checkbox"/> Zeppelin 0.10.1
--	---	---

**Summary** Info

**Name and applications - required**

**Name**  
wine-ml-emr-cc

**Amazon EMR release**  
emr-6.15.0

**Application bundle**  
Spark Interactive (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.7.1, Spark 3.4...)

**Cluster configuration - required**

**Uniform instance groups**  
Primary (m4.large), Core (m4.large)

**Cluster scaling and provisioning - required**

**Provisioning configuration**  
Core size: 3 instances

#### Step 2 : Cluster Configuration:

- Primary Node : Select node type m4.large
- Core Node : Select node type m4.large
- Keep EBS volume as default which is 15 GiB
- Remove Spot node section (Not needed)
- Give core count node as 3 (Total 4 nodes 3 core and 1 master)

#### Snippet:

Amazon EMR > EMR on EC2: Clusters > Create cluster

**Primary**  
Choose EC2 instance type

m4.large  
2 vCore 8 GiB memory  
EBS only 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

m4.large  
2 vCore 8 GiB memory  
EBS only storage On-Demand price: -  
Lowest Spot price: -

Actions

Remove instance group

► Node configuration - optional

Add task instance group

You can add up to 48 more task instance groups.

**EBS root volume**  
EBS root volume applies to the operating systems and applications that you install on the cluster. [EBS root volume ratio constraints](#)

Size (GiB)	IOPS	Throughput (MiB/s)
15	3000	125
15 - 100 GiB per volume General Purpose SSD (gp3)	3000 - 16000 IOPS per volume. Choose a maximum ratio of 50:1 between IOPS	125 - 1000 MiB/s per volume. Choose a maximum ratio of 0.25:1 between throughput

### Step 3 : Networking Requirement & Cluster termination criteria:

- Select the default VPC and subnet in networking.
- Keep cluster termination criteria as default.

### Snippet:

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-0fb28a85a5a200bb7

Browse Create VPC

**Subnet** Info

subnet-00273bf392927cb1a

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

▼ **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.

### Step 4 :Create an EC2 Pair and Amazon EMR Service Role:

- Create a new EC2 pair and save it as .pem file
- Use default EMR\_defaultRole

## Snippet:

▼ 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, authorization, and instance metadata service settings.

Q Choose a security configuration

🔄

Browse [↗](#)

Create security configuration [↗](#)

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

Q wine-ml-emr-cc

✕

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

EMR\_DefaultRole

🔄

EC2 instance profile for Amazon EMR

Name a

Name

wine-ml-cc

Amazon EMR release

emr-6.15.0

Application bundle

Spark Interactive (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.7.1, Spark 3.4....)

Cluster

Uniform instance groups

Primary (m4.large)

Cluster scaling and provisioning - required

Provisioning configuration

Core size: 3 instances

## Step 5 : Select default EC2 instance profile for Amazon EMR:

- Select default Instance profile “EMR\_EC2\_DefaultRole”
- After this select create cluster

## Snippet:

▼ 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

EMR\_DefaultRole

🔄

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

EMR\_EC2\_DefaultRole

🔄

Custom automatic scaling role - **optional**

When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. [Learn more](#)

Custom automatic scaling role

Choose IAM role

🔄

Create IAM role [↗](#)

Cancel

Clone cluster

**Note :** Select only the required field, we can skip the optional setting, after creating cluster will go in creating state for 10 mins then it should go in waiting state, this is when cluster is ready to be used.

This is how cluster will look like once it's available:

Snippet of Cluster Info :

The screenshot shows the Amazon EMR console for a cluster named 'wine-ml-emr-cc'. The cluster is in the 'Starting' state. Key details include: Cluster ID j-LK5153873H5R, Amazon EMR version emr-6.15.0, and a creation time of April 29, 2023, at 23:13 UTC-04:00. The cluster has 1 Primary instance and 3 Core instances. The console also displays sections for Applications (Hadoop 3.3.6, Hive 3.13.3, etc.), Cluster management (Log destination, Persistent application, etc.), and Status and time (Status, Creation time, Elapsed time).

Snippet of Cluster Hardware :

The screenshot shows the Amazon EMR console for the same cluster, focusing on the 'Instances (Hardware)' tab. It displays the instance group settings, including the cluster scaling option set to 'Manually set cluster size'. The instance groups table shows two groups: 'Primary' with 1 instance (ig-M2F5T3G7SGL) and 'Core' with 3 instances (ig-35120TG1QPKE). Both instances are in the 'Running' state.

## 2 : Create an S3 Bucket and upload the required files :

- Created a S3 bucket with Bucket name wine-ml-bucket-vk722
- Put name and basic details and Click create bucket
- Upload following Files to S3

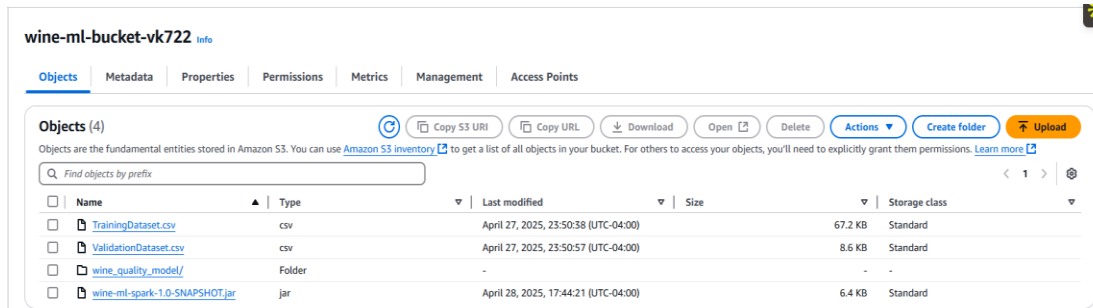
Steps to Upload:

- Select the bucket and click "Upload".
- Drag and drop the files or browse to select them.
- Confirm upload by clicking "Upload".

Uploaded following files:

- **TrainingDataset.csv** – Training dataset
- **ValidationDataset.csv** – Validation Dataset image file

## Snippet of uploaded files:

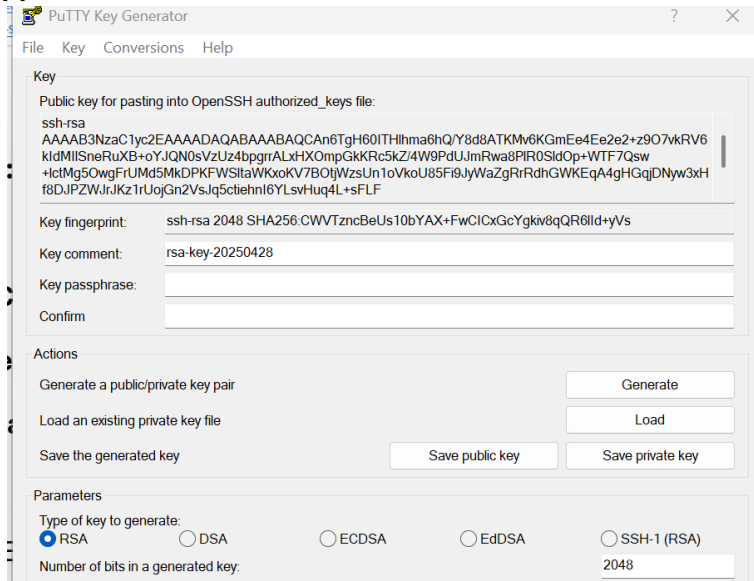


## Initial Setup on Cloud:

### 1. Generate the Private Key(.ppk) using PuttyGen:

- Convert the .pem file to .ppk file using PuttyGen
- Click Load and upload the .pem file of EMR cluster
- Then click on Save Private key, will create .ppk file

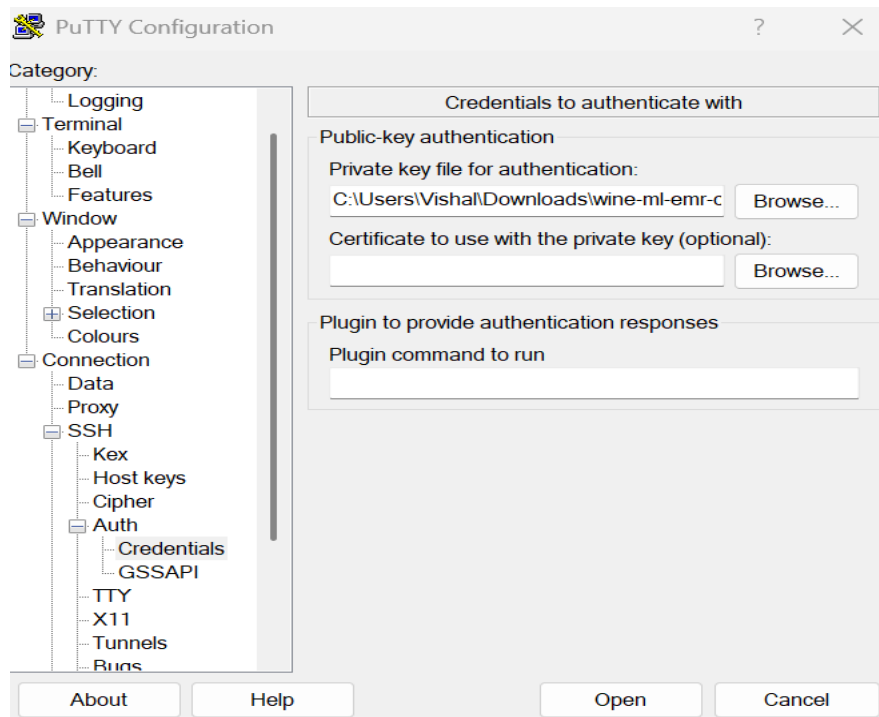
### Snippet:



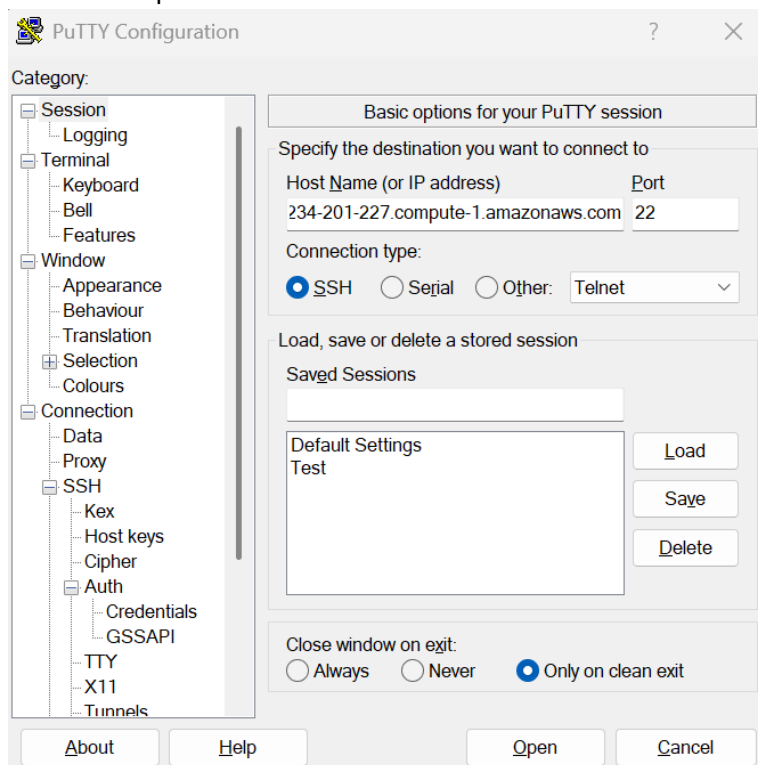
### 2. Connect to EMR Cluster using Putty :

- Click on Connection then SHH then Auth and load .ppk file created in step 1

## Snippet:



- After that enter the hostname of primary node of your EMR cluster in Hostname in putty e.g hostname : [hadoop@ec2-34-234-201-227.compute-1.amazonaws.com](https://hadoop@ec2-34-234-201-227.compute-1.amazonaws.com)
- After that click on open



- After connecting to Cluster, it should look like the snippet below:





### 3. Install necessary things which we need throughout project on EMR cluster:

#### 1. Install git on EMR Cluster & Verify Version:

Run below commands one by one

- `sudo yum update -y`
- `sudo yum install git -y`
- `git --version`

```
Verifying : perl-Git-2.47.1-1.amzn2.0.2.noarch
Verifying : git-2.47.1-1.amzn2.0.2.x86_64
Verifying : lperl-Error-0.17020-2.amzn2.noarch
Verifying : git-core-doc-2.47.1-1.amzn2.0.2.noarch
Verifying : git-core-2.47.1-1.amzn2.0.2.x86_64

Installed:
  git.x86_64 0:2.47.1-1.amzn2.0.2

Dependency Installed:
  git-core.x86_64 0:2.47.1-1.amzn2.0.2  git-core-doc.noarch 0:2.47.1-1.amzn2.0.2  perl-Error.noarch 1:0.17020-2.amzn2  perl-Git.noarch 1:2.47.1-1.amzn2.0.2

Complete!
[hadoop@ip-172-31-58-38 ~]$
[hadoop@ip-172-31-58-38 ~]$ git --version
git version 2.47.1
[hadoop@ip-172-31-58-38 ~]$
[hadoop@ip-172-31-58-38 ~]$
```

#### 2. Install Docker & Verify Version:

Run below commands one by one

- `sudo amazon-linux-extras enable docker`
- `sudo yum install docker -y`
- `sudo service docker start`
- `sudo usermod -aG docker hadoop`
- `sudo docker info`

```
[hadoop@ip-172-31-58-38 CC-CS643-Prgm-Assgn-2-]$ sudo docker info
Client:
Version: 25.0.8
Context: default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version: 0.12.1
    Path: /usr/libexec/docker/cli-plugins/docker-buildx
Server:
Containers: 0
Running: 0
Paused: 0
Stopped: 0
Images: 0
Server Version: 25.0.8
Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
Logging Driver: json-file
Cgroup Driver: cgroupfs
Cgroup Version: 1
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 runc
Default Runtime: runc
Init Binary: docker-init
  containerd version: 05f44ec0a9a75232cd4580270a83437aa3f4da
  runc version: 6c52b3fc541fb26fe6c374d5f581120a5dbda6e
  init version: de40ad0
Security Options:
  seccomp
    Profile: builtin
Kernel Version: 4.14.355-275.603.amzn2.x86_64
Operating System: Amazon Linux 2
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 7.775GiB
Name: ip-172-31-58-38
ID: 9455685f-7c2d-4987-bdb5-36c1abdae06f
Docker Root Dir: /mnt/var/lib/docker
Debug Mode: false
Experimental: false
Insecure Registries:
  127.0.0.0/8
```

### 3. Login to Docker Hub:

- Run below command and enter the Docker hub credentials:  
command : `sudo docker login`

```
Username: vishalk722
Password:
WARNING! Your password will be stored unencrypted in /home/hadoop/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[hadoop@ip-172-31-58-38 CC-CS643-Prgm-Assgn-2-]$
```

## Running the Project-Step by Step Guide:

### 1. Now Clone the Git Hub Repo on EMR Cluster:

- `git clone https://github.com/vishal2609/CC-CS643-Prgm-Assgn-2-.git`

```
[hadoop@ip-172-31-63-75 ~]$ git clone https://github.com/vishal2609/CC-CS643-Prgm-Assgn-2-.git
Cloning into 'CC-CS643-Prgm-Assgn-2-'...
remote: Enumerating objects: 103, done.
remote: Counting objects: 100% (103/103), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 103 (delta 22), reused 102 (delta 21), pack-reused 0 (from 0)
Receiving objects: 100% (103/103), 649.66 KiB | 19.69 MiB/s, done.
Resolving deltas: 100% (22/22), done.
[hadoop@ip-172-31-63-75 ~]$
```

### 2. Give permission to the folder :

- `chmod 777 CC-CS643-Prgm-Assgn-2/`

```

[hadoop@ip-172-31-63-75 ~]$ cd CC-CS643-Prgm-Assgn-2-
[hadoop@ip-172-31-63-75 ~]$ chmod 777 CC-CS643-Prgm-Assgn-2-/
[hadoop@ip-172-31-63-75 ~]$ ls
CC-CS643-Prgm-Assgn-2-
[hadoop@ip-172-31-63-75 ~]$

```

### 3. Change directory to CC-CS643-Prgm-Assgn-2-

- cd CC-CS643-Prgm-Assgn-2-/

```

[hadoop@ip-172-31-63-75 ~]$ cd CC-CS643-Prgm-Assgn-2-/
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ ls
DOCKERFILE pom.xml README.md screenshots src target
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$

```

### 4. Submitting Spark Job For Parallel Training

#### Command:

```

spark-submit \
--class com.wine.WinePrediction \
--master yarn \
--deploy-mode cluster \
--num-executors 4 \
--executor-cores 2 \
--executor-memory 2G \
target/wine-ml-spark-1.0-SNAPSHOT.jar

```

#### Snippet:

```

[hadoop@ip-172-31-48-14 CC-CS643-Prgm-Assgn-2-]$ spark-submit \
> --class com.wine.WinePrediction \
> --master yarn \
> --deploy-mode cluster \
> --num-executors 4 \
> --executor-cores 2 \
> --executor-memory 2G \
> target/wine-ml-spark-1.0-SNAPSHOT.jar
25/04/30 21:39:22 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/04/30 21:39:23 INFO DefaultHARMFalloverProxyProvider: Connecting to ResourceManager at ip-172-31-48-14.ec2.internal/172.31.48.14:8032
25/04/30 21:39:24 INFO Configuration: resource-types.xml not found
25/04/30 21:39:24 INFO ResourceUtils: Unable to find 'resource-types.xml'.
25/04/30 21:39:24 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (6144 MB per container)
25/04/30 21:39:24 INFO Client: Will allocate AM container, with 2432 MB memory including 384 MB overhead
25/04/30 21:39:24 INFO Client: Setting up container launch context for our AM
25/04/30 21:39:24 INFO Client: Setting up the launch environment for our AM container
25/04/30 21:39:24 INFO Client: Preparing resources for our AM container
25/04/30 21:39:24 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
25/04/30 21:41:05 INFO Client: Application report for application_1746042755223_0004 (state: RUNNING)
25/04/30 21:41:06 INFO Client: Application report for application_1746042755223_0004 (state: RUNNING)
25/04/30 21:41:07 INFO Client: Application report for application_1746042755223_0004 (state: RUNNING)
25/04/30 21:41:08 INFO Client: Application report for application_1746042755223_0004 (state: RUNNING)
25/04/30 21:41:09 INFO Client: Application report for application_1746042755223_0004 (state: RUNNING)
25/04/30 21:41:10 INFO Client: Application report for application_1746042755223_0004 (state: FINISHED)
25/04/30 21:41:10 INFO Client:
client token: N/A
diagnostics: N/A
ApplicationMaster host: ip-172-31-59-217.ec2.internal
ApplicationMaster RPC port: 44331
queue: default
start time: 1746049174307
final status: SUCCEEDED
tracking URL: http://ip-172-31-48-14.ec2.internal:20888/proxy/application_1746042755223_0004/
user: hadoop
25/04/30 21:41:10 INFO ShutdownHookManager: Shutdown hook called
25/04/30 21:41:10 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-bb3b1a75-7dbc-4e38-96f7-bddaf728d82d
25/04/30 21:41:10 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-b5edc5b2-6d58-4f81-b5f3-5a4c57f97afd

```

## 5. Confirm the model is saved on s3:

Snippet:

wine-ml-bucket-vk722 [Info](#)

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

**Objects (4)** [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

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 objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">TrainingDataset.csv</a>	csv	April 27, 2025, 23:50:38 (UTC-04:00)	67.2 KB	Standard
<input type="checkbox"/>	<a href="#">ValidationDataset.csv</a>	csv	April 27, 2025, 23:50:57 (UTC-04:00)	8.6 KB	Standard
<input type="checkbox"/>	<a href="#">wine_quality_model</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">wine-ml-spark-1.0-SNAPSHOT.jar</a>	jar	April 28, 2025, 17:44:21 (UTC-04:00)	6.4 KB	Standard

## Yarn log of stating job ran on 4 nodes in parallel setup – 3 Core and 1 Master

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources
1	0	0	1	0	<memory 0 B, vCores 0>	<memory 18 GB, vCores 12>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy
3	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit-M), vcores]	<memory 1, vCores 1>	<memory 6144, vCores 4>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU V-Cores	Allocated Memory MB
application_1746042755223_0001	hadoop	Wine Quality Prediction	SPARK		default	0	Wed Apr 30 16:11:20 -0400 2025	Wed Apr 30 16:11:22 -0400 2025	Wed Apr 30 16:12:37 -0400 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A

Showing 1 to 1 of 1 entries

## 6. Running Prediction without Docker on EMR cluster:

Command:

```
spark-submit \
  --class com.wine.WinePredictApp \
  --master yarn \
  target/wine-ml-spark-1.0-SNAPSHOT.jar
```

Snippet:

```
[hadoop@ip-172-31-48-14 CC-CS643-Prgm-Assgn-2-J$ spark-submit \
> --class com.wine.WinePredictApp \
> --master yarn \
> target/wine-ml-spark-1.0-SNAPSHOT.jar

25/04/30 20:19:26 INFO SparkContext: Running Spark version 3.4.1-amzn-2
25/04/30 20:19:27 INFO ResourceUtils: =====
25/04/30 20:19:27 INFO ResourceUtils: No custom resources configured for spark.driver.
25/04/30 20:19:27 INFO ResourceUtils: =====
25/04/30 20:19:27 INFO SparkContext: Submitted application: Wine Quality Prediction - Prediction Phase
25/04/30 20:19:27 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: ,
Heap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
25/04/30 20:19:27 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
25/04/30 20:19:27 INFO ResourceProfileManager: Added ResourceProfile id: 0
25/04/30 20:19:27 INFO SecurityManager: Changing view acls to: hadoop
25/04/30 20:19:27 INFO SecurityManager: Changing modify acls to: hadoop
25/04/30 20:19:27 INFO SecurityManager: Changing view acls groups to:
25/04/30 20:19:27 INFO SecurityManager: Changing modify acls groups to:
25/04/30 20:19:27 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: hadoop; gr
: groups with modify permissions: EMPTY
```

## Confirm the prediction:

```
25/04/30 20:20:36 INFO BlockManagerInfo: Added broadcast_12_piece0 in memory on ip-172-31-48-14.ec2.internal:38483 (size: 2.9 KiB, f
25/04/30 20:20:36 INFO SparkContext: Created broadcast_12 from broadcast at DAGScheduler.scala:1592
25/04/30 20:20:36 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 6 (ShuffledRDD[26] at reduceByKey at MulticlassMetri
25/04/30 20:20:36 INFO YarnScheduler: Adding task set 6.0 with 1 tasks resource profile 0
25/04/30 20:20:36 INFO TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6) (ip-172-31-50-248.ec2.internal, executor 1, partition 0
25/04/30 20:20:36 INFO BlockManagerInfo: Added broadcast_12_piece0 in memory on ip-172-31-50-248.ec2.internal:44967 (size: 2.9 KiB, f
25/04/30 20:20:36 INFO MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to 172.31.50.248:39014
25/04/30 20:20:37 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 273 ms on ip-172-31-50-248.ec2.internal (executor 1)
25/04/30 20:20:37 INFO YarnScheduler: Removed TaskSet 6.0, whose tasks have all completed, from pool
25/04/30 20:20:37 INFO DAGScheduler: ResultStage 6 (collectAsMap at MulticlassMetrics.scala:61) finished in 0.347 s
25/04/30 20:20:37 INFO DAGScheduler: Job 5 is finished. Cancelling potential speculative or zombie tasks for this job
25/04/30 20:20:37 INFO YarnScheduler: Killing all running tasks in stage 6: Stage finished
25/04/30 20:20:37 INFO DAGScheduler: Job 5 finished: collectAsMap at MulticlassMetrics.scala:61, took 4.865467 s
F1 Score on Validation Data = 0.5625522927084354
25/04/30 20:20:37 INFO SparkContext: SparkContext is stopping with exitCode 0.
25/04/30 20:20:37 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-48-14.ec2.internal:4040
25/04/30 20:20:37 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/04/30 20:20:37 INFO YarnClientSchedulerBackend: Shutting down all executors
25/04/30 20:20:37 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/04/30 20:20:37 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/04/30 20:20:37 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/30 20:20:37 INFO MemoryStore: MemoryStore cleared
25/04/30 20:20:37 INFO BlockManager: BlockManager stopped
25/04/30 20:20:37 INFO BlockManagerMaster: BlockManagerMaster stopped
25/04/30 20:20:37 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/04/30 20:20:37 INFO SparkContext: Successfully stopped SparkContext
25/04/30 20:20:37 INFO ShutdownHookManager: Shutdown hook called
25/04/30 20:20:37 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-14fd4e97-cda8-462a-9d51-38695b8971e8
```

## 7. Running Prediction without Docker on Single instance:

### Check the EC2 running instance of Cluster :

#### Snippet:

Instances (4) <a href="#">Info</a>									
Find Instance by attribute or tag (case-sensitive)									
All states									
Instance state = running X Clear filters									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
<input type="checkbox"/>		i-0466d4206739a74c5	Running	m4.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1e	ec2-18-209-179-5	
<input type="checkbox"/>		i-0eadd2c835bc56102	Running	m4.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1e	ec2-54-157-148-1	
<input type="checkbox"/>		i-06d7a2460dc038f2	Running	m4.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1e	ec2-100-25-33-11	
<input type="checkbox"/>		i-099e821570c09965e	Running	m4.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1e	ec2-100-26-158-1	

### Connect to your master EC2 instance:

#### Connect to instance [Info](#)

Connect to your instance i-099e821570c09965e using any of these options

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
<b>Instance ID</b> i-099e821570c09965e			
<ol style="list-style-type: none"><li>Open an SSH client.</li><li>Locate your private key file. The key used to launch this instance is wine-ml-emr-cc.pem</li><li>Run this command, if necessary, to ensure your key is not publicly viewable. chmod 400 "wine-ml-emr-cc.pem"</li><li>Connect to your instance using its Public DNS: ec2-100-26-158-165.compute-1.amazonaws.com</li></ol>			
<b>Example:</b> ssh -i "wine-ml-emr-cc.pem" root@ec2-100-26-158-165.compute-1.amazonaws.com			
<b>Note:</b> In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.			

### Go to folder where the pem file is stored and open cmd & run the command to connect:

- ssh -i "wine-ml-emr-cc.pem" ec2-user@ec2-100-26-158-165.compute-1.amazonaws.com

#### Snippet:

### Snippet:

```
[hadoop@ip-172-31-48-14 CC-CS643-Prgm-Assgn-2]$ aws s3 ls s3://wine-ml-bucket-vk722/
PRE wine_quality_model/
2025-04-28 03:50:38      68804 TrainingDataset.csv
2025-04-28 03:50:57      8760 ValidationDataset.csv
2025-04-28 21:44:21      6541 wine-ml-spark-1.0-SNAPSHOT.jar
[hadoop@ip-172-31-48-14 CC-CS643-Prgm-Assgn-2]$ spark-submit \
> --class com.wine.WinePredictApp \
> --master yarn \
> target/wine-ml-spark-1.0-SNAPSHOT.jar

25/04/30 21:02:56 INFO SparkContext: Running Spark version 3.4.1-amzn-2
25/04/30 21:02:56 INFO ResourceUtils: =====
25/04/30 21:02:56 INFO ResourceUtils: No custom resources configured for spark.driver.
25/04/30 21:02:56 INFO ResourceUtils: =====
25/04/30 21:02:56 INFO SparkContext: Submitted application: Wine Quality Prediction - Prediction Phase
25/04/30 21:02:56 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4,
t: 4269, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, am
25/04/30 21:02:56 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
25/04/30 21:02:56 INFO ResourceProfileManager: Added ResourceProfile id: 0
25/04/30 21:02:56 INFO SecurityManager: Changing view acls to: hadoop
25/04/30 21:02:56 INFO SecurityManager: Changing modify acls to: hadoop
25/04/30 21:02:56 INFO SecurityManager: Changing view acls groups to:
25/04/30 21:02:56 INFO SecurityManager: Changing modify acls groups to:
25/04/30 21:02:56 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions:
with modify permissions: hadoop; groups with modify permissions: EMPTY
25/04/30 21:02:56 INFO Utils: Successfully started service 'sparkDriver' on port 46067.
25/04/30 21:02:57 INFO SparkEnv: Registering MapOutputTracker
25/04/30 21:02:57 INFO SparkEnv: Registering BlockManagerMaster
25/04/30 21:02:57 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology info
25/04/30 21:02:57 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint un
```

### Confirm the prediction:

```
s Vector(0))
25/04/30 21:03:58 INFO YarnScheduler: Adding task set 6.0 with 1 tasks resource profile 0
25/04/30 21:03:58 INFO TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6) (ip-172-31-50-248.ec2.internal, executor 2, 6
25/04/30 21:03:58 INFO BlockManagerInfo: Added broadcast_12_piece0 in memory on ip-172-31-50-248.ec2.internal:43791 (size:
25/04/30 21:03:58 INFO MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to 172.31.50.248:51
25/04/30 21:03:58 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 166 ms on ip-172-31-50-248.ec2.internal (e
25/04/30 21:03:58 INFO YarnScheduler: Removed TaskSet 6.0, whose tasks have all completed, from pool
25/04/30 21:03:58 INFO DAGScheduler: ResultStage 6 (collectAsMap at MulticlassMetrics.scala:61) finished in 0.189 s
25/04/30 21:03:58 INFO DAGScheduler: Job 5 is finished. Cancelling potential speculative or zombie tasks for this job
25/04/30 21:03:58 INFO YarnScheduler: Killing all running tasks in stage 6: Stage finished
25/04/30 21:03:58 INFO DAGScheduler: Job 5 finished: collectAsMap at MulticlassMetrics.scala:61, took 1.808258 s
F1 Score on Validation Data = 0.5625522927084354
25/04/30 21:03:58 INFO SparkContext: SparkContext is stopping with exitCode 0.
25/04/30 21:03:58 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-48-14.ec2.internal:4040
25/04/30 21:03:58 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/04/30 21:03:58 INFO YarnClientSchedulerBackend: Shutting down all executors
25/04/30 21:03:58 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/04/30 21:03:58 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/04/30 21:03:58 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/30 21:03:58 INFO MemoryStore: MemoryStore cleared
25/04/30 21:03:58 INFO BlockManager: BlockManager stopped
25/04/30 21:03:58 INFO BlockManagerMaster: BlockManagerMaster stopped
```

## Now to Run Prediction with Docker, Lets set up Docker Setup:

### 1. Build Docker Image:

Command: `sudo docker build -t wine-predictor:v3 .`

#### Snippet:

```
=>=> naming to docker.io/library/wine-predictor:v3
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2]$ sudo docker build -t wine-predictor:v3 .
[+] Building 0.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 385B
=> [internal] load metadata for docker.io/bitnami/spark:3.3.0
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/bitnami/spark:3.3.0@sha256:b4e81939f1606b1f039bealb86145317ae38c5803a6b6895b38961a4b1213c7c
=> [internal] load build context
=> => transferring context: 85B
=> CACHED [2/3] WORKDIR /app
=> CACHED [3/3] COPY target/wine-ml-spark-1.0-SNAPSHOT.jar /app/app.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:05a344e7d1ebb4274c3ef18daa0b488ffcdcd68181ff5d14af0113b3921102f
=> => naming to docker.io/library/wine-predictor:v3
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2]$
```

### Verify Docker Image

- `sudo docker images`

```
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ ^C
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
wine-predictor       v1              05a344e7d1eb   About a minute ago   1.25GB
wine-predictor       v3              05a344e7d1eb   About a minute ago   1.25GB
vishalk722/wine-ml-app v2             6bbeea84c4dc   2 hours ago        1.25GB
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$
```

## 2. Now Tag the Docker Images:

- `sudo docker tag wine-predictor:v3 vishalk722/wine-ml-app:v3`

```
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ sudo docker tag wine-predictor:v3 vishalk722/wine-ml-app:v3
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
wine-predictor       v1              05a344e7d1eb   4 minutes ago   1.25GB
wine-predictor       v3              05a344e7d1eb   4 minutes ago   1.25GB
vishalk722/wine-ml-app v3             05a344e7d1eb   4 minutes ago   1.25GB
vishalk722/wine-ml-app v2             6bbeea84c4dc   2 hours ago     1.25GB
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$
```

## 3. Push the docker Image to Docker Hub

- `sudo docker push vishalk722/wine-ml-app:v3`

```
Upload an image to a registry
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$ sudo docker push vishalk722/wine-ml-app:v3
The push refers to repository [docker.io/vishalk722/wine-ml-app]
81d37d4f1692: Pushed
7643bc492431: Layer already exists
c0c1eb24d971: Layer already exists
877c59cba308: Layer already exists
v3: digest: sha256:a7383743acdf89ff1e850423d6d65df00a1e5a18d15e09a1aefb852224d0f06 size: 1157
[hadoop@ip-172-31-63-75 CC-CS643-Prgm-Assgn-2-]$
```

Verify on Docker Hub if the image pushed:

Snippet:

☐ Sort by Newest

TAG

v3

Last pushed 1 minute by [vishalk722](#)

Digest

[a7383743acdf](#)

OS/ARCH

linux/amd64

Last pull

less than 1 day

Compressed size

722.38 MB

docker pull vishalk722/wine-ml-app:v3

Copy

TAG

v2

Last pushed about 1 hour by [vishalk722](#)

Digest

[446cb3e5e2ef](#)

OS/ARCH

linux/amd64

Last pull

less than 1 day

Compressed size

722.38 MB

docker pull vishalk722/wine-ml-app:v2

Copy

TAG

v1

Last pushed about 2 hours by [vishalk722](#)

docker pull vishalk722/wine-ml-app:v1

Copy

## 4. Now clean local images on EMR :

- `sudo docker rmi -f IMAGE_ID`
- E.g: `sudo docker rmi -f 6bbeea84c4dc`



## Snippet:

```
[hadoop@ip-172-31-63-75 ~]$ sudo docker rmi -f 05a344e7d1eb
Untagged: wine-predictor:v1
Untagged: wine-predictor:v3
Untagged: vishalk722/wine-ml-app:v3
Untagged: vishalk722/wine-ml-app@sha256:a7383743acdff89ff1e850423d6d65df00ae1e5a18d15e09a1aefb852224d0f06
Deleted: sha256:05a344e7d1ebb4274c3ef18daa0b488ffcd68181ff5d14af0113b3921102f
[hadoop@ip-172-31-63-75 ~]$ sudo docker rmi -f 6bbeea84c4dc
Untagged: vishalk722/wine-ml-app:v2
Untagged: vishalk722/wine-ml-app@sha256:44dcb3e5e2ef23d8f9561224080c6a2c1e183690d9268dddc8e97377201b1212
Deleted: sha256:6bbeea84c4dc3f74af2bcfc6b522e6b99556442d7e954af83a3032fde47f07cb
[hadoop@ip-172-31-63-75 ~]$ sudo docker rmi -f 05a344e7d1eb
Error response from daemon: No such image: 05a344e7d1eb:latest
[hadoop@ip-172-31-63-75 ~]$ sudo docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
[hadoop@ip-172-31-63-75 ~]$
```

## 5. Remove clone repo directory : CC-CS643-Prgm-Assgn-2-/

- `rm -r CC-CS643-Prgm-Assgn-2/`


## Snippet:

```
[hadoop@ip-172-31-63-75 ~]$ rm -r CC-CS643-Prgm-Assgn-2-/
rm: remove write-protected regular file 'CC-CS643-Prgm-Assgn-2-/.git/objects/pack/pack-cb920cd430970530d3e80b66f94e9339b1a657c7.pack'? y
rm: remove write-protected regular file 'CC-CS643-Prgm-Assgn-2-/.git/objects/pack/pack-cb920cd430970530d3e80b66f94e9339b1a657c7.rev'? y
rm: remove write-protected regular file 'CC-CS643-Prgm-Assgn-2-/.git/objects/pack/pack-cb920cd430970530d3e80b66f94e9339b1a657c7.idx'? y
[hadoop@ip-172-31-63-75 ~]$ ls
[hadoop@ip-172-31-63-75 ~]$
```


## 6. Now Pull Image from docker hub and verify its available:


- Copy command from docker Hub :
- `docker pull vishalk722/wine-ml-app:v16`

## Snippet:

**vishalk722/wine-ml-app** 

Last pushed about 7 hours ago • Repository size: 1.6 GB

[Add a description](#) 

[Add a category](#) 

General

**Tags**

Image Management BETA

Collaborators

Webhooks

Settings


☐

Sort by


Newest

Delete

TAG

 **v16**

Last pushed about 7 hours by [vishalk722](#)

Digest	OS/ARCH	Last pull	Compressed size 
<a href="#">58e8c2e0e5ee</a>	linux/amd64	less than 1 day	755.93 MB

docker pull vishalk722/wine-ml-app:v16

Copy

Docker commands

To push a new tag to this repository:

```
docker push vishalk722/wine-ml-app:tagname
```

## Run the command on EMR Cluster

```
[hadoop@ip-172-31-52-39 ~]$ docker pull vishalk722/wine-ml-app:v16
v16: Pulling from vishalk722/wine-ml-app
001c52e26ad5: Already exists
d9dab5b6e964: Already exists
2068746027ec: Already exists
9dae329d350: Already exists
d85151f15b66: Already exists
52a8c426d30b: Already exists
8754a66e0050: Already exists
bddc7b093fe0: Already exists
987e953de113: Already exists
3db0f77b3231: Already exists
d2f7dd03f9e: Already exists
c632c60fa4e8: Already exists
4c2bf67b1d55: Already exists
Digest: sha256:58e8cce0e5ee3e368115a757a4360e233b6f65912806439b2677650350a48850
Status: Downloaded newer image for vishalk722/wine-ml-app:v16
docker.io/vishalk722/wine-ml-app:v16
[hadoop@ip-172-31-52-39 ~]$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
vishalk722/wine-ml-app  v16                9430a7864634       12 minutes ago     1.15GB
```

## 7. Running Prediction with Docker image on EMR Cluster:

**Command:** `sudo docker run --rm -v $HOME/.ivy2:/root/.ivy2 -v $HOME:/root -e HOME=/root -e SPARK_SUBMIT_OPTS="-Ddivy.cache.dir=/root/.ivy2/cache -Ddivy.home=/root/.ivy2" --user root vishalk722/wine-ml-app:v16`

**Snippet:**

```
[hadoop@ip-172-31-52-39 ~]$ sudo docker run --rm -v $HOME/.ivy2:/root/.ivy2 -v $HOME:/root -e HOME=/root -e SPARK_SUBMIT_OPTS="-Ddivy.cache.dir=/root/.ivy2/cache -Ddivy.home=/root/.ivy2" --user root vishalk722/wine-ml-app:v16
25/04/30 12:14:57 INFO SparkContext: Running Spark version 3.3.3
25/04/30 12:14:57 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/04/30 12:14:57 INFO ResourceUtils: =====
25/04/30 12:14:57 INFO ResourceUtils: No custom resources configured for spark.driver.
25/04/30 12:14:57 INFO ResourceUtils: =====
25/04/30 12:14:57 INFO SparkContext: Submitted application: Wine Quality Prediction - Prediction Phase
25/04/30 12:14:57 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , heap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpu -> name: cpu, amount: 1.0)
25/04/30 12:14:57 INFO ResourceProfile: Limiting resource is cpu
25/04/30 12:14:57 INFO ResourceProfileManager: Added ResourceProfile id: 0
25/04/30 12:14:57 INFO SecurityManager: Changing view acls to: root
25/04/30 12:14:57 INFO SecurityManager: Changing modify acls to: root
25/04/30 12:14:57 INFO SecurityManager: Changing view acls groups to:
25/04/30 12:14:57 INFO SecurityManager: Changing modify acls groups to:
25/04/30 12:14:57 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
```

## 8. Confirm the result with F1 score printed on validation data.

**Snippet:**

```
25/04/30 12:14:31 INFO MemoryStore: Block broadcast_11_piece0 stored as bytes in memory (estimated size 2.8 KiB, free 365.8 MiB)
25/04/30 12:14:31 INFO BlockManagerInfo: Added broadcast_11_piece0 in memory on 61b7169ae867:44583 (size: 2.8 KiB, free: 366.2 MiB)
25/04/30 12:14:31 INFO SparkContext: Created broadcast_11 from broadcast at DAGScheduler.scala:1509
25/04/30 12:14:31 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 6 (ShuffledRDD[25] at reduceByKey at MulticlassMetrics.scala:61)
25/04/30 12:14:31 INFO TaskSchedulerImpl: Adding task set 6.0 with 1 tasks resource profile 0
25/04/30 12:14:31 INFO TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6) (61b7169ae867, executor driver, partition 0, NODE_LOCAL, 4096)
25/04/30 12:14:31 INFO Executor: Running task 0.0 in stage 6.0 (TID 6)
25/04/30 12:14:31 INFO ShuffleBlockFetcherIterator: Getting 1 (490.0 B) non-empty blocks including 1 (490.0 B) local and 0 (0.0 B) host-local blocks
25/04/30 12:14:31 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 14 ms
25/04/30 12:14:31 INFO Executor: Finished task 0.0 in stage 6.0 (TID 6). 2162 bytes result sent to driver
25/04/30 12:14:31 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 88 ms on 61b7169ae867 (executor driver) (1/1)
25/04/30 12:14:31 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
25/04/30 12:14:31 INFO DAGScheduler: ResultStage 6 (collectAsMap at MulticlassMetrics.scala:61) finished in 0.110 s
25/04/30 12:14:31 INFO DAGScheduler: Job 5 is finished. Cancelling potential speculative or zombie tasks for this job
25/04/30 12:14:31 INFO TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
25/04/30 12:14:31 INFO DAGScheduler: Job 5 finished: collectAsMap at MulticlassMetrics.scala:61, took 1.413798 s
F1 Score on Validation Data = 0.5625522927084354
25/04/30 12:14:31 INFO SparkUI: Stopped Spark web UI at http://61b7169ae867:4040
25/04/30 12:14:31 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/30 12:14:31 INFO MemoryStore: MemoryStore cleared
25/04/30 12:14:31 INFO BlockManager: BlockManager stopped
25/04/30 12:14:31 INFO BlockManagerMaster: BlockManagerMaster stopped
25/04/30 12:14:31 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/04/30 12:14:31 INFO SparkContext: Successfully stopped SparkContext
25/04/30 12:14:31 INFO ShutdownHookManager: Shutdown hook called
25/04/30 12:14:31 INFO ShutdownHookManager: Deleting directory /tmp/spark-c14c0a41-ade7-4409-b263-66ddbf15127b
25/04/30 12:14:31 INFO ShutdownHookManager: Deleting directory /tmp/spark-f295550c-59af-488c-b30c-6205d577478e
25/04/30 12:14:31 INFO MetricsSystemImpl: Stopping s3a-file-system metrics system...
```

## 9. Running Prediction with Docker image single cluster:

- Pull docker image from Docker Hub on primary EC2 which we connected on CMD  
**Command :** `docker pull vishalk722/wine-ml-app:v16`

**Snippet:**

```

Login Succeeded
[hadoop@ip-172-31-48-14 ~]$ sudo docker pull vishalk722/wine-ml-app:v16
v16: Pulling from vishalk722/wine-ml-app
001c52e26ad5: Extracting [>] 557.1kB/55MB
d9d4b9b6e964: Download complete
2068746827ec: Download complete
9daef329d350: Download complete
d85151f15b66: Download complete
52a8c426d30b: Download complete
8754a66e0050: Downloading [=====] 39.39MB/105.9MB
bddc7b093fe0: Download complete
987e953de113: Waiting
3db0f77b3231: Waiting
d2f77dd03f9e: Waiting
c632c60fa4e8: Waiting
4c2bf67b1d55: Waiting

```

**Now run the docker image on cluster :**

**Command :** `sudo docker run --rm -v $HOME/.ivy2:/root/.ivy2 -v $HOME:/root -e HOME=/root -e SPARK_SUBMIT_OPTS="-Divy.cache.dir=/root/.ivy2/cache -Divy.home=/root/.ivy2" --user root vishalk722/wine-ml-app:v16`

**Snippet:**

```

docker.io/vishalk722/wine-ml-app:v16
[hadoop@ip-172-31-48-14 ~]$ sudo docker run --rm -v $HOME/.ivy2:/root/.ivy2 -v $HOME:/root -e HOME=/root -e SPARK_SUBMIT_OPTS="-Divy.cache.dir=/root/.ivy2/cache -Divy.home=/root/.ivy2" --user root vishalk722/wine-ml-app:v16

```

```

$ vector(0)
25/04/30 21:34:38 INFO TaskSchedulerImpl: Adding task set 6.0 with 1 tasks resource profile 0
25/04/30 21:34:38 INFO TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6) (602a44af77ce, executor driver, partition 0, NODE
25/04/30 21:34:38 INFO Executor: Running task 0.0 in stage 6.0 (TID 6)
25/04/30 21:34:38 INFO ShuffleBlockFetcherIterator: Getting 1 (490.0 B) non-empty blocks including 1 (490.0 B) local and 0 (0.0
(0.0 B) remote blocks
25/04/30 21:34:38 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 32 ms
25/04/30 21:34:38 INFO Executor: Finished task 0.0 in stage 6.0 (TID 6). 2162 bytes result sent to driver
25/04/30 21:34:38 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 115 ms on 602a44af77ce (executor driver) (1/1)
25/04/30 21:34:38 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
25/04/30 21:34:38 INFO DAGScheduler: ResultStage 6 (collectAsMap at MulticlassMetrics.scala:61) finished in 0.140 s
25/04/30 21:34:38 INFO TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
25/04/30 21:34:38 INFO DAGScheduler: Job 5 finished: collectAsMap at MulticlassMetrics.scala:61, took 1.382377 s
F1 Score on Validation Data = 0.5625522927084354
25/04/30 21:34:38 INFO SparkUI: Stopped Spark web UI at http://602a44af77ce:4040
25/04/30 21:34:38 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/04/30 21:34:38 INFO MemoryStore: MemoryStore cleared
25/04/30 21:34:38 INFO BlockManager: BlockManager stopped
25/04/30 21:34:38 INFO BlockManagerMaster: BlockManagerMaster stopped
25/04/30 21:34:38 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/04/30 21:34:38 INFO SparkContext: Successfully stopped SparkContext
25/04/30 21:34:38 INFO ShutdownHookManager: Shutdown hook called
25/04/30 21:34:38 INFO ShutdownHookManager: Deleting directory /tmp/spark-#brc#02fa-76b6-#f91-b2ad-a03c9756925c

```