**Introduction**:

The purpose of this report is to provide a detailed description of the development process of the wine quality prediction ML model in Spark over AWS. This report will cover the dataset used, the model implementation, the Docker container setup, and the instructions to set up and run the cloud environment for model training and application prediction.

**Dataset**:

Two datasets were provided for this assignment - TrainingDataset.csv and ValidationDataset.csv. The TrainingDataset.csv was used to train the model in parallel on multiple EC2 instances, while the ValidationDataset.csv was used to validate the model and optimize its performance. The TestDataset.csv was not provided, but a file with a similar structure was used to test the functionality and performance of the prediction application.

**Model Implementation:**

The model was implemented using a Spark application that used MLlib to train for wine quality prediction using the training dataset. The validation dataset was used to check the performance of the trained model and to potentially tune the ML model parameters for best performance. A simple linear regression or logistic regression model from MLlib was used initially, and multiple ML models were tried to see which one leads to better performance. For classification models, 10 classes were used (the wine scores are from 1 to 10). The F1 score was used as the measure of prediction performance, which is available in MLlib.

**Cloud Environment Setup:**

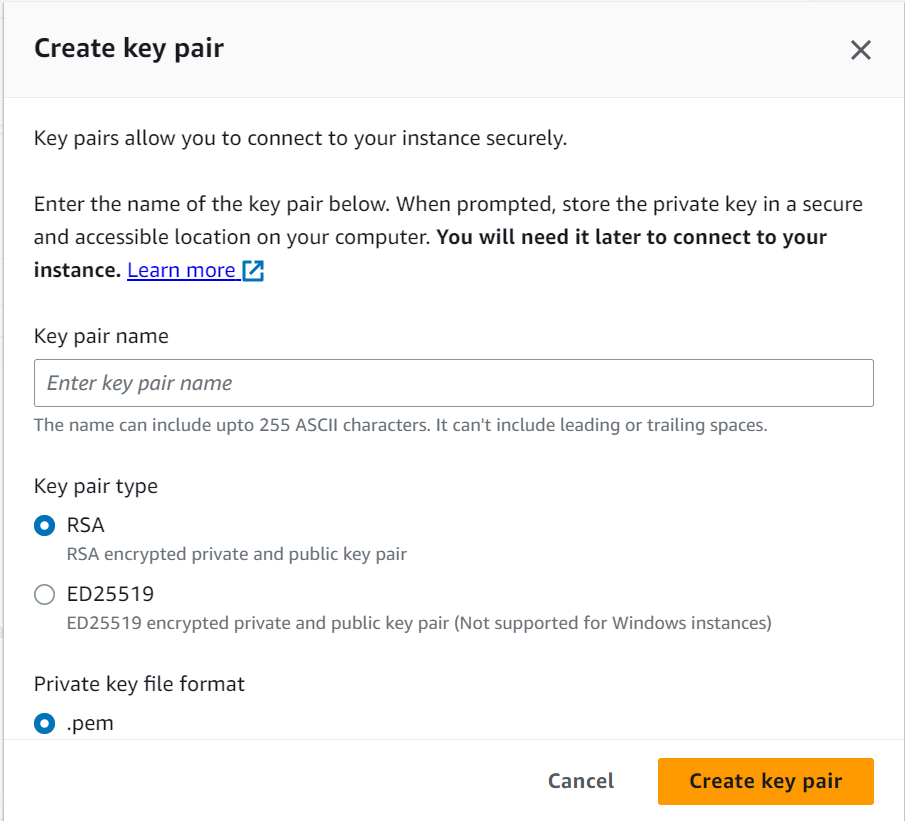
To set up the cloud environment for model training and application prediction, the following steps were taken:

1. Graphical user interface, diagram

   Description automatically generatedSign in using AWS Credentials by going on to the link https://console.aws.amazon.com/ec2/

1. Graphical user interface, text, application

   Description automatically generatedGo to EC2 dashboard and click on Launch Instance
2. Graphical user interface, application

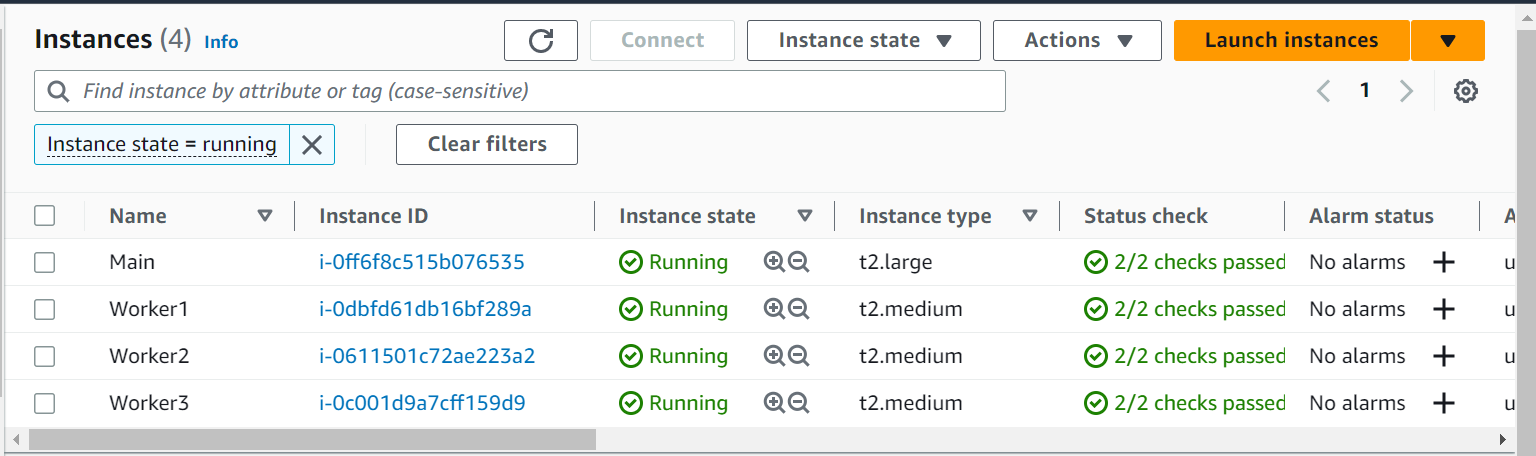
   Description automatically generatedIn Name enter a name, we will use “worker”.
3. In Application and OS Images , under Quick Start , select Ubuntu
4. Under Key pair ( login ) , create a new Key pair and save it in your PC.

Graphical user interface, text, application, email

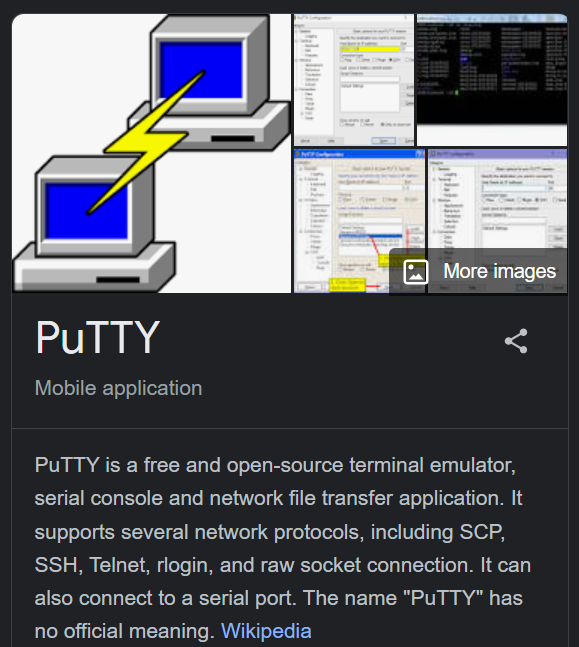
Description automatically generated

1. Under Summary make the number of instances to 4.

Four EC2 instances were launched in the AWS cloud platform.

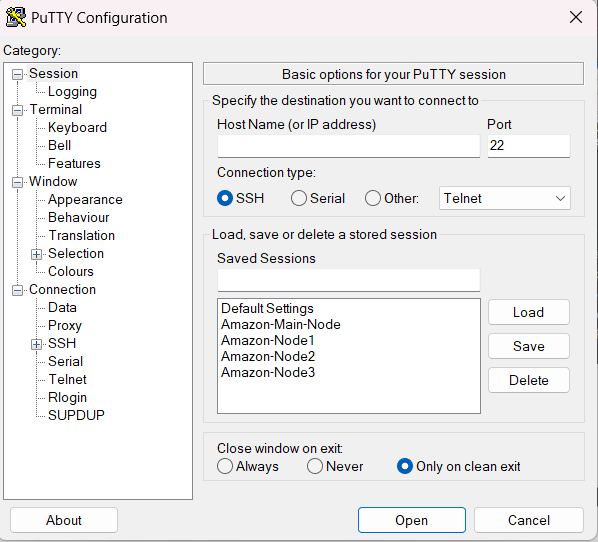
  
  
Please rename the instances as per your convenience.

Now we will configure each Ec2 instance to setup our spark cluster. Now download the following two software



WinSCP PuTTY

Now open PuTTy and configure four profiles for easy SSH to the EC2 instances.



We have made four profiles

**Amazon-Main-Node**

**Amazon-Node1**

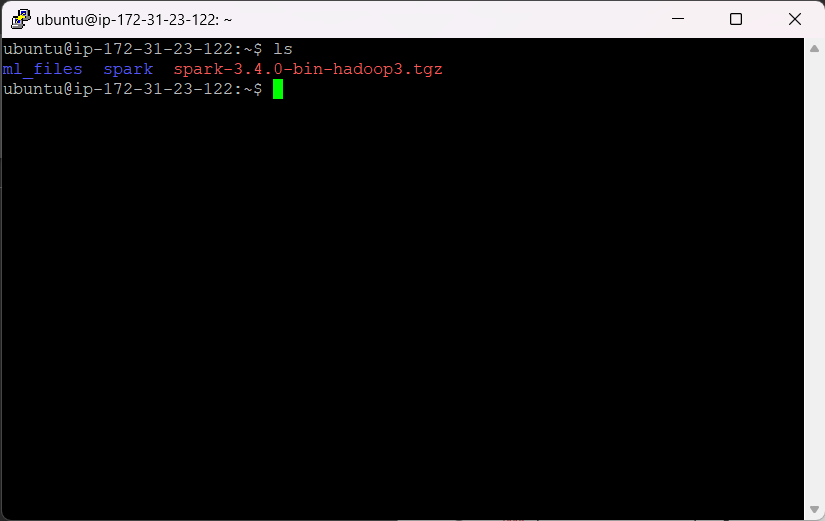
**Amazon-Node2**

**Amazon-Node3**

Now we will SSH into each machine and install spark using the following commands,

sudo apt-get update && sudo apt-get upgrade- y  
wget <https://dlcdn.apache.org/spark/spark-3.4.0/spark-3.4.0-bin-hadoop3.tgz>  
tar -xvf spark-3.4.0-bin-hadoop3.tgz  
mkdir spark ml\_files  
cd spark-3.4.0-bin-hadoop3 && cp \* ../spark/ -r  
rm -rf spark-3.4.0-bin-hadoop3

The home directory would look like this

  
  
Now lets configure the environment variables , enter the following command

sudo nano ~/.bashrc

Add the following lines at the end of the file

Text

Description automatically generated

Save and exit.

Now run the following command

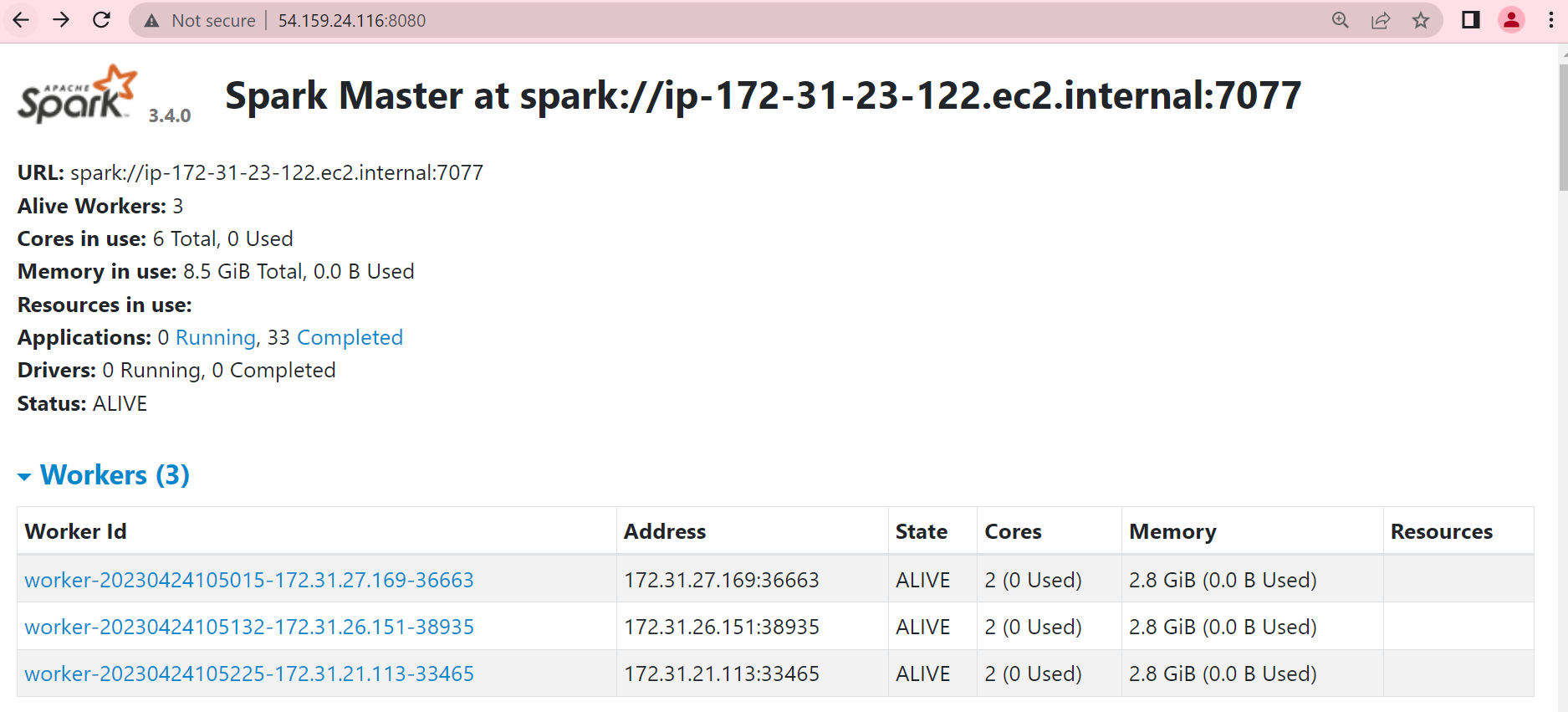
source ~/.bashrc

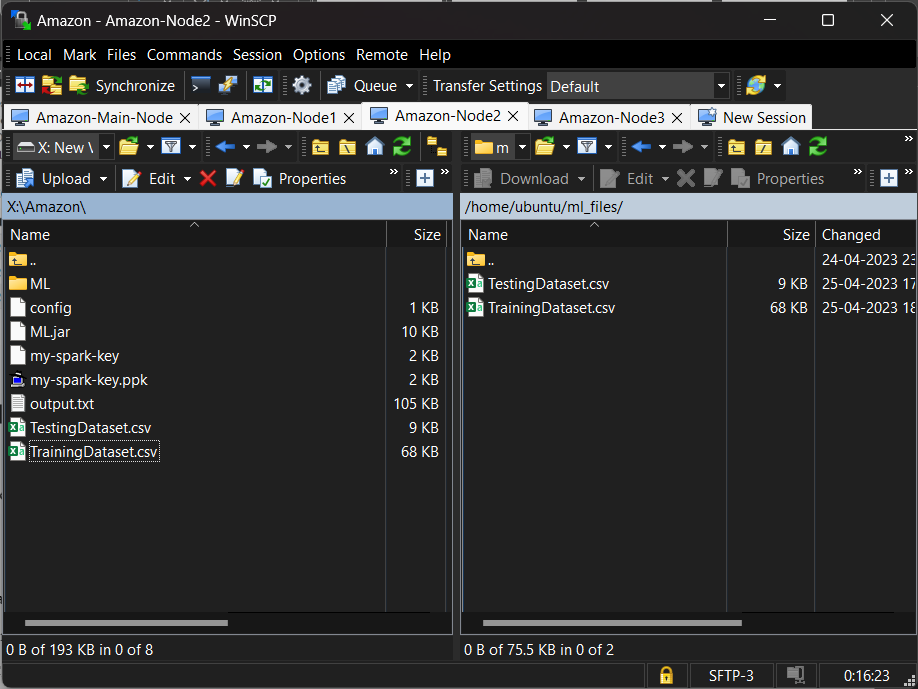
Repeat the above same steps on all four EC2 instance and run the following command in Master Node

./sbin/start-slave.sh

And run the following command in the Worker Nodes

./sbin/start-slave.sh spark://ip-172-31-23-122.ec2.internal:7077

You will see the following output when you visit the public IP of your master node at port 8080.  


Now use WinSCP and copy the TrainingDataset.csv file and TestingDataset.csv file in the directory shown below:  
  
  
  
Perform the same for all Nodes.

Now using place the jar file in the folder ml\_lib in the master node and run the following command.

spark-submit --class JavaClassify --master spark://ip-172-31-23-122.ec2.internal:7077 --executor-memory 2G --total-executor-cores 6 /home/ubuntu/spark/ml\_files/ML.jar

It will train and save the ML learning Model in the directory.

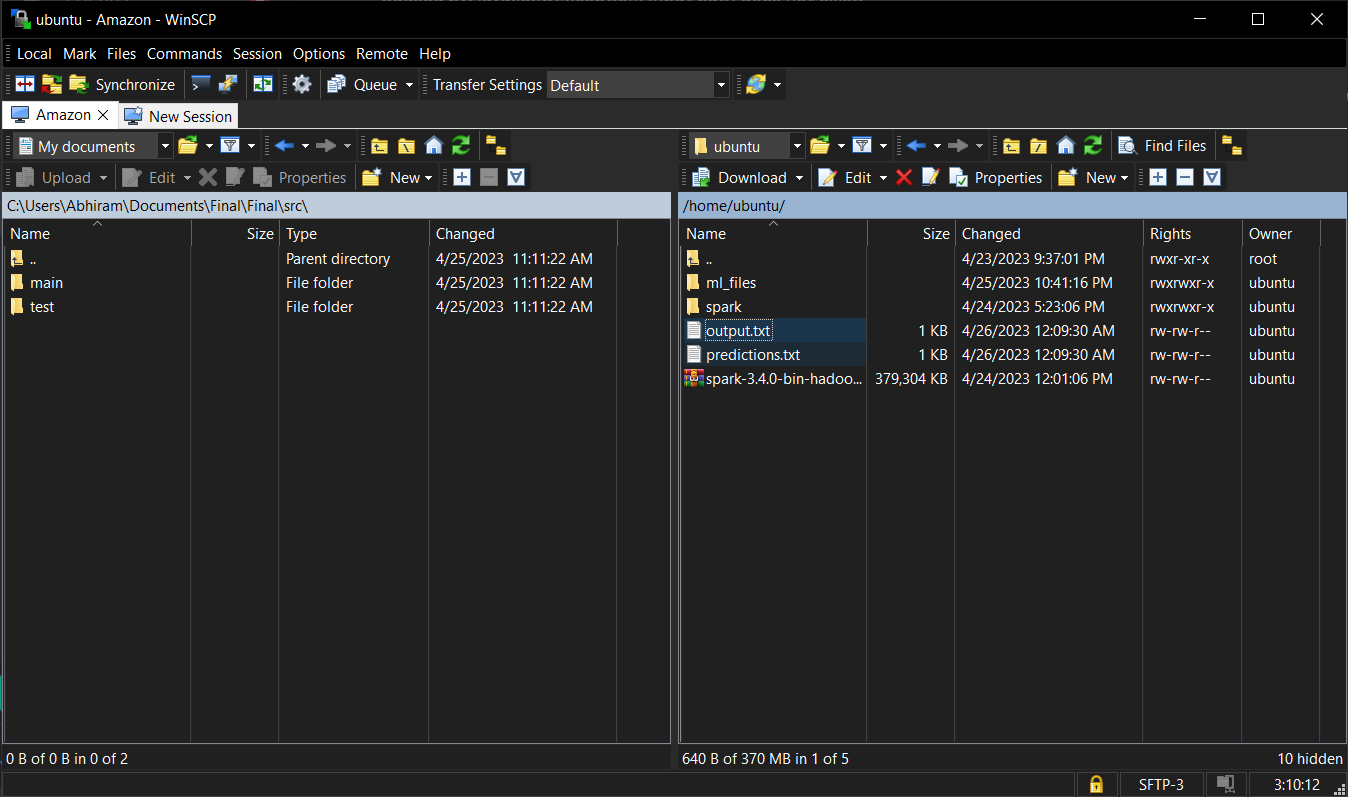
**Prediction Application**:

In order to run the prediction application, use the following command:

spark-submit --class PredictionApp --master spark://ip-172-31-23-122.ec2.internal:7077 --executor-memory 2G --total-executor-cores 6 /home/ubuntu/spark/ml\_files/ML.jar TestingDataset.csv

The last argument passed is the dataset to be evaluated.

The output will be saved in a file named output.txt



A screenshot of a computer

Description automatically generated with medium confidence

It contains the predicted values for the given file.

**Conclusion**:

In conclusion, this report has provided a detailed description of the development process of the wine quality prediction ML model in Spark over AWS.