1. Create an EC2 Key Pair:”  
   A screenshot of a computer

   Description automatically generated  
   A screenshot of a computer

   Description automatically generated
2. Give it a name and select .pem if you have MAC OS or .ppk if you have Windows and click on “Create Key pair”  
   A screenshot of a computer

   Description automatically generated
3. Once Created it should appear in the list:  
   A screenshot of a computer

   Description automatically generated
4. Create the EMR Cluster:  
   A screenshot of a computer

   Description automatically generated
5. Keep these as it is:  
   A screenshot of a computer

   Description automatically generated
6. Select Core instances as 1 and Task-1 instances as 3. So total 4 instances for training.

A screenshot of a computer

Description automatically generated

1. Select the Cluster termination as per your convenience. I had selected 3 hours (my lucky number)  
   A screenshot of a computer

   Description automatically generated
2. Keep these as it is  
   A screenshot of a computer

   Description automatically generated
3. Now select the key that you have created earlier as shown below  
   A screenshot of a computer program

   Description automatically generated
4. Select the EMR\_DefaultRole as the service role and EMR\_EC2\_DefaultRole as the instance profile and the rest as it is.

A screenshot of a computer

Description automatically generated

1. Verify the summary and create the cluster.  
   A screenshot of a computer

   Description automatically generated
2. Go Amazon S3 and create an S3 bucket with a name as shown below  
   A screenshot of a computer

   Description automatically generated
3. Select the AWS region, give a name to the S3 bucket, keep the rest settings as it is  
   A screenshot of a bucket

   Description automatically generated  
   A screenshot of a computer

   Description automatically generated
4. Once created it should appear in the list of s3 buckets in the AWS Management Console  
   A screenshot of a computer

   Description automatically generated
5. Once created the bucket click on it and upload the TrainingDataset.csv and ValidationDataset.csv and the python scripts to train and create the model and to run the created model in src as shown below  
     
   A screenshot of a computer

   Description automatically generated  
   A screenshot of a computer

   Description automatically generated
6. The folder “src/” would have the code that trains and creates the model.
7. Now create a session with the EMR EC2 master node of the WinePredCluster1 using WinSCP. You need to use the winepred.ppk key that you have created in the beginning.  
   A screenshot of a computer screen

   Description automatically generated
8. Connect to the EMR EC2 Master node through putty using the winepred.ppk file from your local as shown below:  
   A computer screen shot of a computer

   Description automatically generated  
   A screenshot of a computer

   Description automatically generated
9. Use ec2-user as the login user

A screen shot of a computer

Description automatically generated

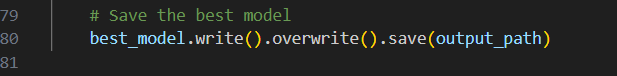
1. After successful you should see something like this  
   A screenshot of a computer

   Description automatically generated
2. Now run the command “sudo spark-submit s3://sk3374-winepred/src/wine\_prediction\_train.py to start and train the model  
   A screen shot of a computer

   Description automatically generated
3. You will see something like this it the code is running and the model is getting trained:  
   A screen shot of a computer

   Description automatically generated  
   A black screen with white text

   Description automatically generated
4. Once the code completed executed it should create a model and store in in the AWS S3 as per our login in the code:  
     
   A computer screen with text and red line

   Description automatically generated  
     
    

From the s3 bucket: As shown below the model would be created in the s3 bucket in the folder trained.model/  
A screenshot of a computer

Description automatically generated

1. Now can you can get the model from your s3 bucket to your EC2 master node using the below command  
   “aws s3 sync s3://sk3374-winepred/trained.model ./trained.model/”  
   A black screen with a white rectangular object

   Description automatically generated  
   Now when you do ls you can see the downloaded model into your instance  
   A screen shot of a computer program

   Description automatically generated
2. Now using WinSCP connect to the master EC2 instance to the script that tests that prediction model and use a sample testdata.csv created using the previous training data or the validation data just to test the model  
   A screenshot of a computer screen

   Description automatically generated
3. Now download the trained.model folder from the EC2 instance to your local and transfer the testdata.csv and wine\_prediction\_test.py (the script to test the accuracy of the trained model) to the EC2 instance  
   A screenshot of a computer

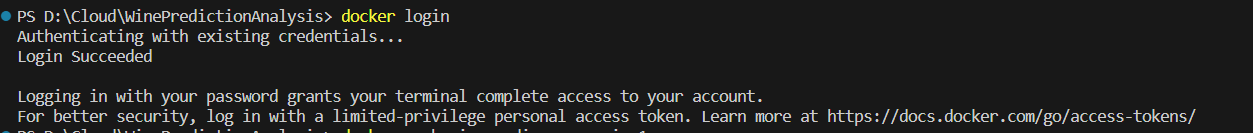
   Description automatically generated
4. One done make sure all the files needed for testing the trained model are on your EC2 instance.  
   A black screen with a white background

   Description automatically generated
5. Now go ahead and run the wine\_prediction\_test.py using the below command  
   python wine\_prediction\_test.py  
   A screen shot of a computer

   Description automatically generated
6. Also you can run the wine\_prediction\_test.py in your local once you get the trained.model from the S3 as shown below  
   A black screen with white and blue squares

   Description automatically generated with medium confidence
7. Now create a Dockerfile with all the required configurations to run the wine\_prediction\_test.py
8. Now build an image using Dockerfile. Below is the command that is used create and build a docker image
9. “docker build -t winepredimage:version1 .”  
   A screenshot of a computer program

   Description automatically generated
10. Once test it using the docker run command as shown below  
    A screenshot of a computer

    Description automatically generated
11. Now use docker login to login to your DockerHub account  
    
12. Now go to DockerHub on a web browser and create a repository called wine-prediction  
    A screenshot of a web browser

    Description automatically generated
13. Now tag the local image to your repository that you have created using the below command  
    “docker tag winepredimage:version1 karthikkk999/wine-prediction:version1”
14. And then push the docker image to the DockerHub using the command  
    “docker push karthikkk999/wine-prediction:version1”
15. You can see this on DockerHub  
    A screenshot of a computer

    Description automatically generated
16. ssh into any of the EC2 instances that you want to run the docker image and install docker  
      
      
      
    “sudo yum install docker”  
    A screenshot of a computer

    Description automatically generated  
    A computer screen with white text

    Description automatically generated
17. Start docker and pull the wine-prediction image that you have built  
    “sudo service docker start”  
    “sudo docker pull karthikkk999/wine-prediction:version1”  
    A screen shot of a computer

    Description automatically generated
18. Once the docker image is pulled run it using the docker run command  
    “sudo docker run karthikkk999/wine-prediction:version1”  
    A screenshot of a computer screen

    Description automatically generated  
      
    DONE!