AWS Spark Wine Quality Prediction Application by Hemanjali Bogadi

FesmaAWSWinePredictionPysparkApp is an application developed with Python's pySpark interface and installed on an AWS EMR cluster. The objective of this project is to simultaneously train a machine learning model on EC2 instances to predict wine quality using publically accessible data, and then utilize the trained model to make wine quality predictions. In order to make deployments easier, the project also uses Docker to produce a container image for a trained machine learning model.

Link to GitHub code - https://github.com/Hema542/AWSSPARKMLAPK.git

Link to docker container image:
https://hub.docker.com/layers/hemanjali693/fesmaawssparkwinepredictionapp/fesmawine-quality-prediction/images/sha256ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687a1fefc?context=repo

Source files Description

fesmawine_prediction.py - reads Training dataset from S3 and trains model in parallel on EMR spark cluster. Once model is trained, it can be run on provided test data provided via the S3 bucket. This program stores trained model in S3 bucket.

fesmawine_test_data_prediction.py - program loads trained model and executes that model on given test data file. This will then print F1 score as metrics for accuracy of the trained model.

Dockerfile - creates docker image and run container for easy deployment.

Create Spark cluster in AWS

User can create spark cluster using EMR console provided by AWS.

Steps to create one with 4 ec2 instance:

- Create Key-Pair for EMR cluster using navigation ```EC2-> Network & Security -> Key-pairs```. Use .pem as format. This will download {name of key pair}>.pem file. Keep it safe you will need that to do SSH to EC2 instances.
- Navigate to Amazon EMR console. Then, navigate to clusters-> create cluster.
- Now fill in respective sections:
  ```
  ```

General Configuratin -> Cluster Name

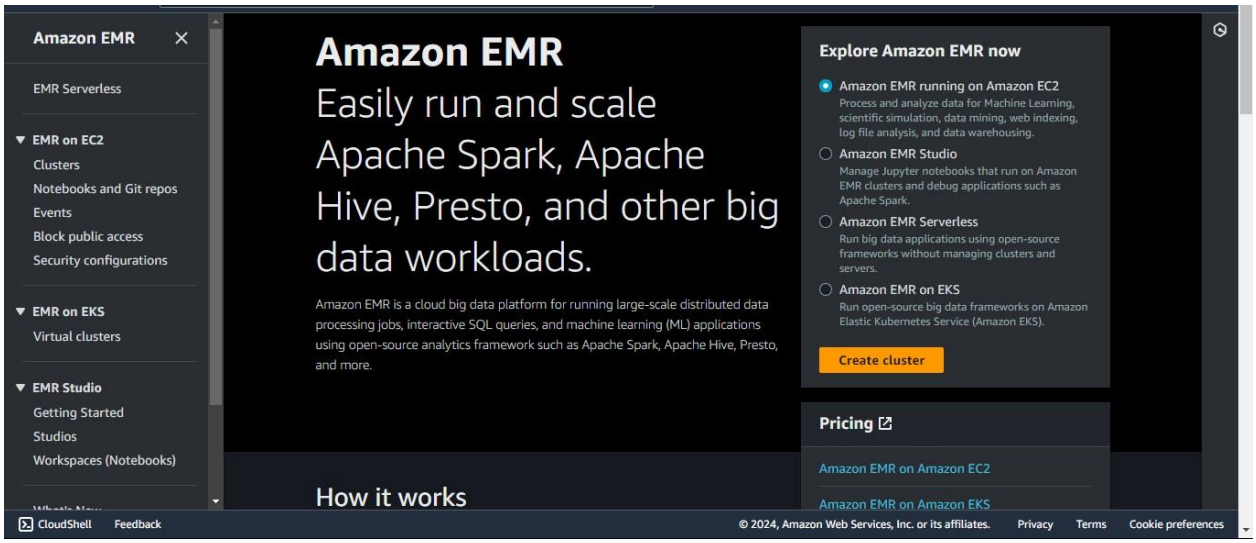
Software Configuration-> EMR 5.33 , do select 'Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.9.0' option menu.

Harware Configuration -> Make instance count as 4

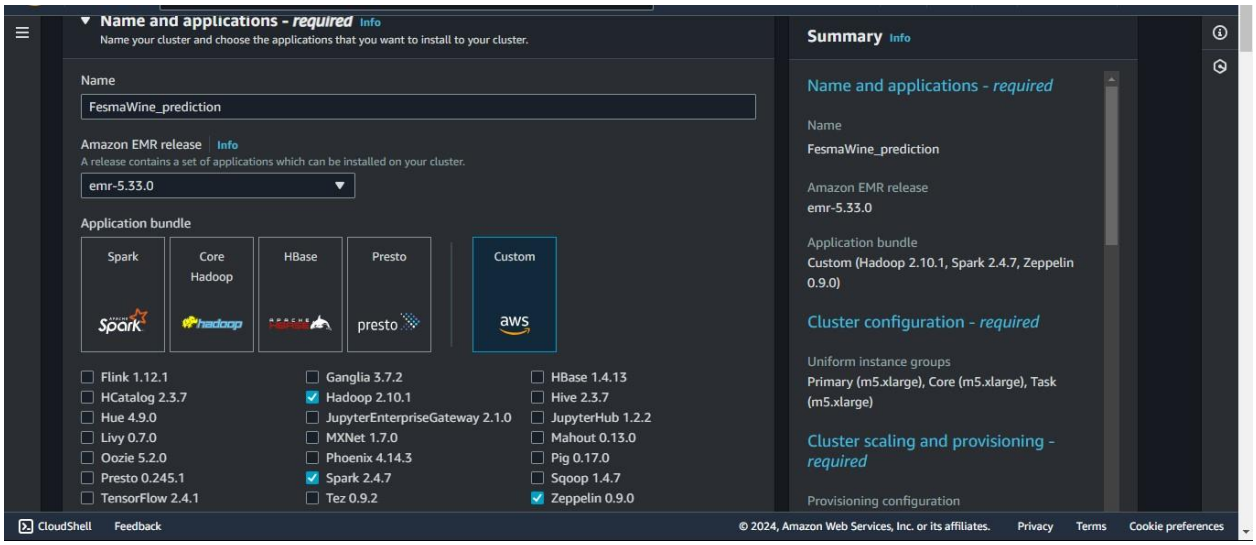
Security Access -> Provide .pem key created in above step.

Rest of parameters can be left default.

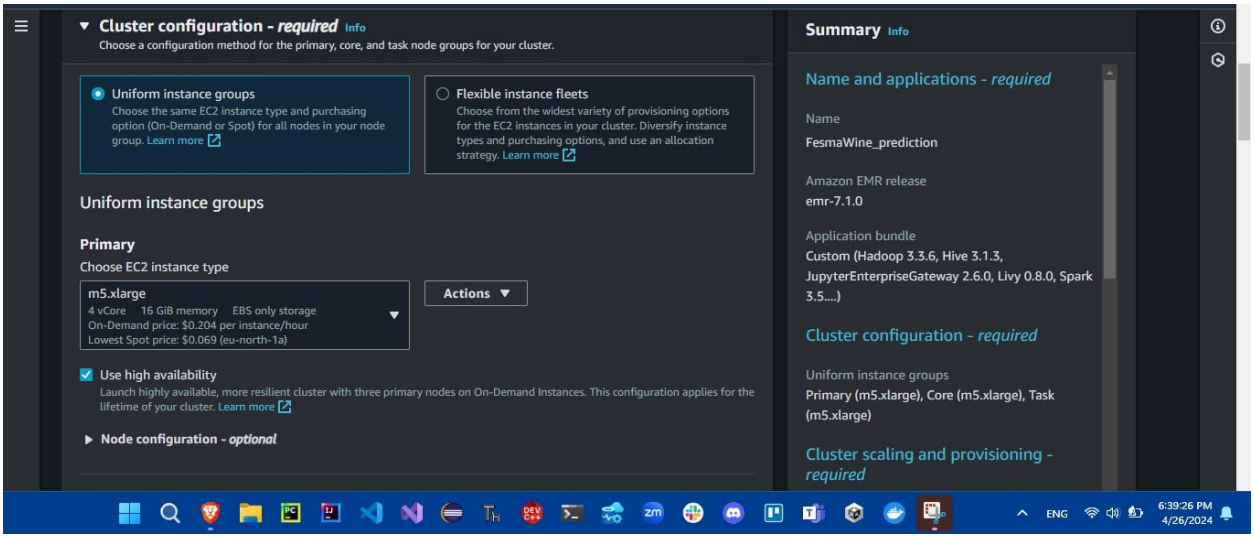
- Cluster status should be 'Waiting' on successful cluster creation

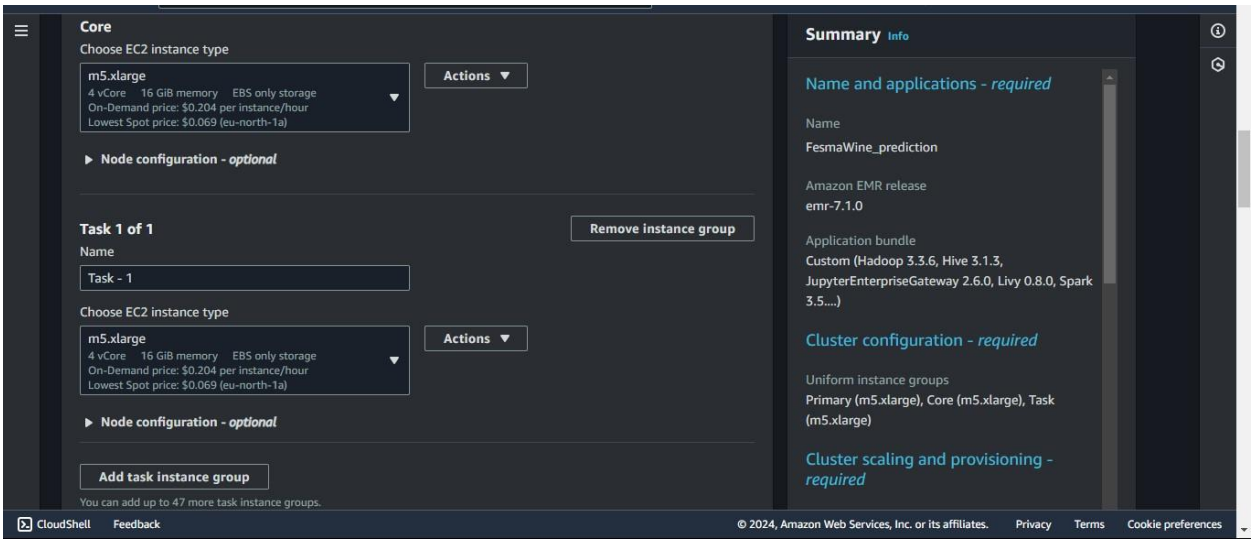


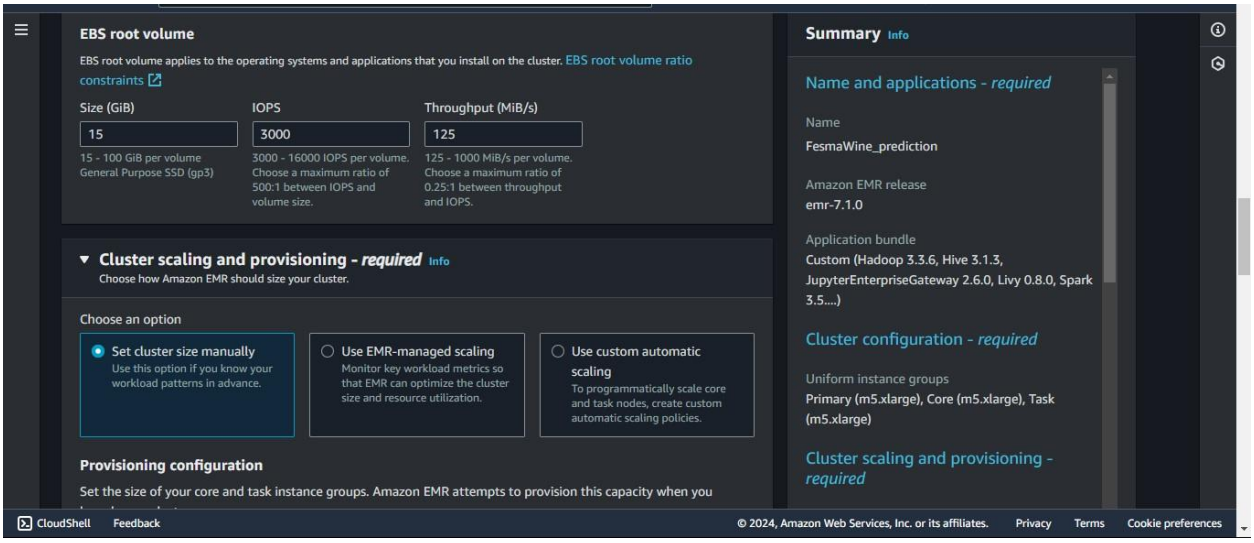
The choose the framework of choice and in this case we will configure our cluster to use Spark 2.4.7 on Hadoop 2.10.1 and Zeppelin 0.90.

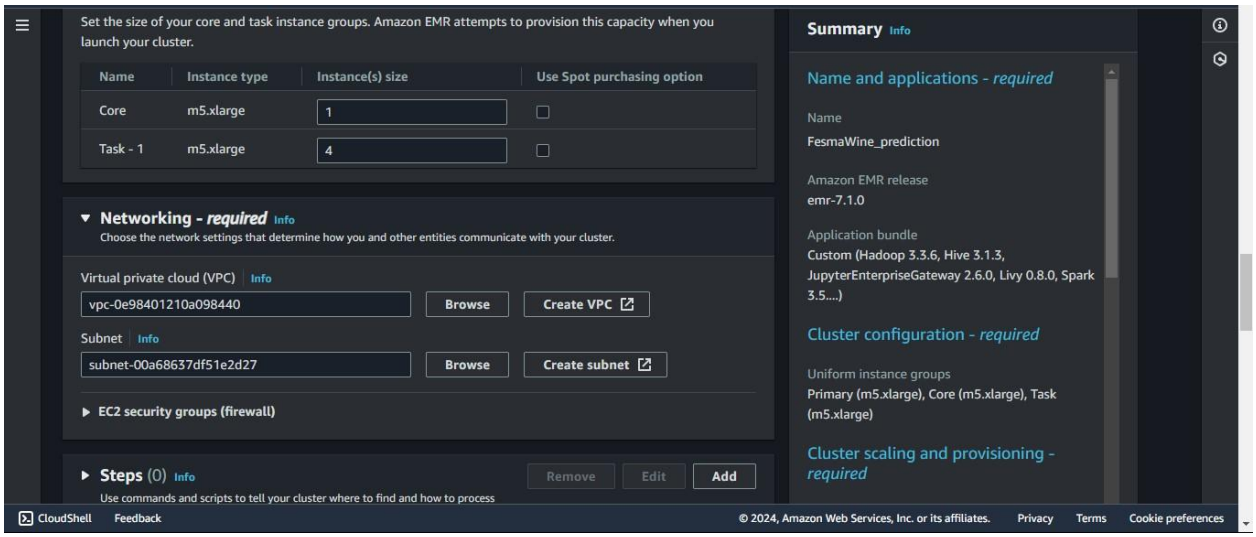


After that, we make the EMR to have same instance groups for the application.

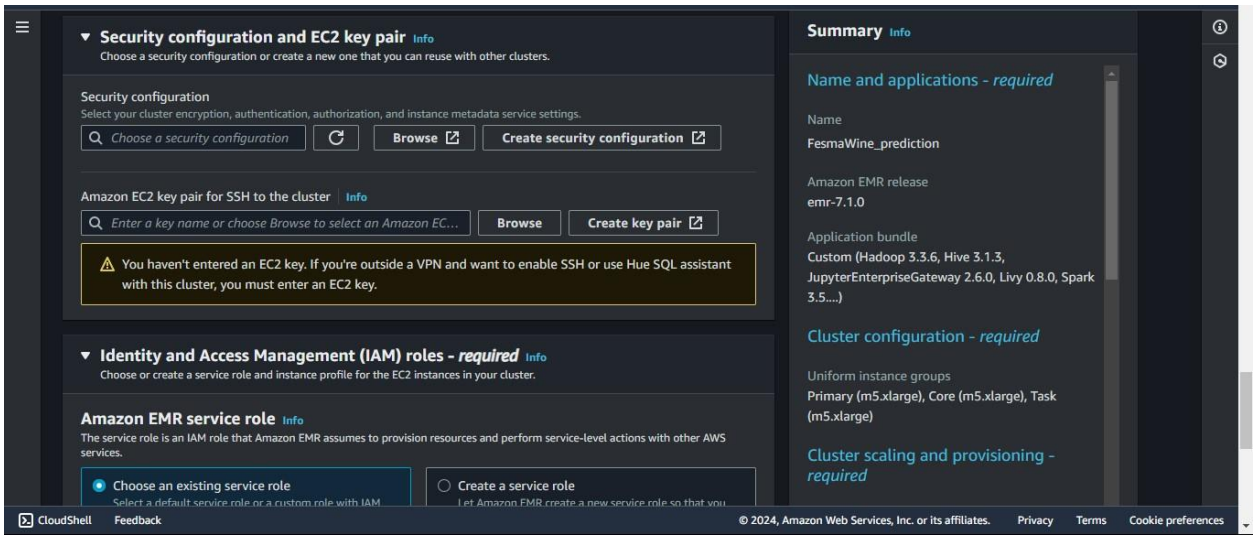

Cluster configuration - required Info
Choose a configuration method for the primary, core, and task node groups for your cluster.
Uniform instance groups
Choose the same EC2 instance type and purchasing option (On-Demand or Spot) for all nodes in your node group. Learn more
Flexible instance fleets
Choose 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
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: $0.204 per instance/hour
Lowest Spot price: $0.069 (eu-north-1a)
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
Summary Info
Name and applications - required
Name
FesmaWine_prediction
Amazon EMR release
emr-7.1.0
Application bundle
Custom (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5....)
Cluster configuration - required
Uniform instance groups
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)
Cluster scaling and provisioning - required
ENG
6:39:26 PM
4/26/2024

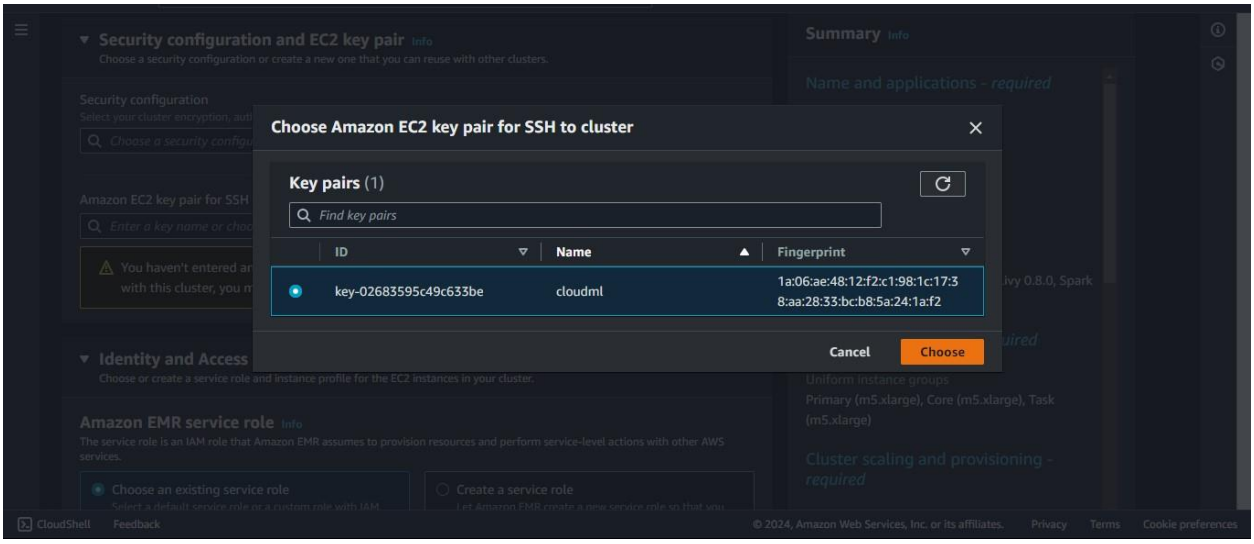

Core
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: $0.204 per instance/hour
Lowest Spot price: $0.069 (eu-north-1a)
Actions
Node configuration - optional
Task 1 of 1
Remove instance group
Name
Task - 1
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: $0.204 per instance/hour
Lowest Spot price: $0.069 (eu-north-1a)
Actions
Node configuration - optional
Add task instance group
You can add up to 47 more task instance groups.
Summary Info
Name and applications - required
Name
FesmaWine_prediction
Amazon EMR release
emr-7.1.0
Application bundle
Custom (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5....)
Cluster configuration - required
Uniform instance groups
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)
Cluster scaling and provisioning - required
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

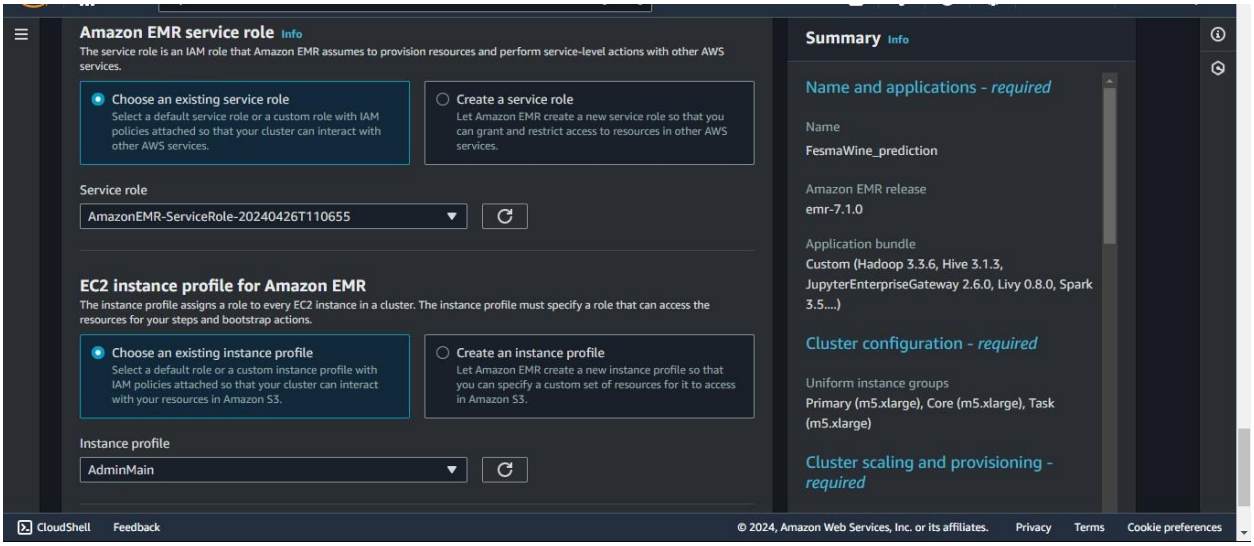

EBS root volume
EBS root volume applies to the operating systems and applications that you install on the cluster. EBS root volume ratio constraints
Size (GiB)
15
15 - 100 GiB per volume
General Purpose SSD (gp3)
IOPS
3000
3000 - 16000 IOPS per volume. Choose a maximum ratio of 500:1 between IOPS and volume size.
Throughput (MiB/s)
125
125 - 1000 MiB/s per volume. Choose a maximum ratio of 0.25:1 between throughput and IOPS.
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
Summary Info
Name and applications - required
Name
FesmaWine_prediction
Amazon EMR release
emr-7.1.0
Application bundle
Custom (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5....)
Cluster configuration - required
Uniform instance groups
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)
Cluster scaling and provisioning - required
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



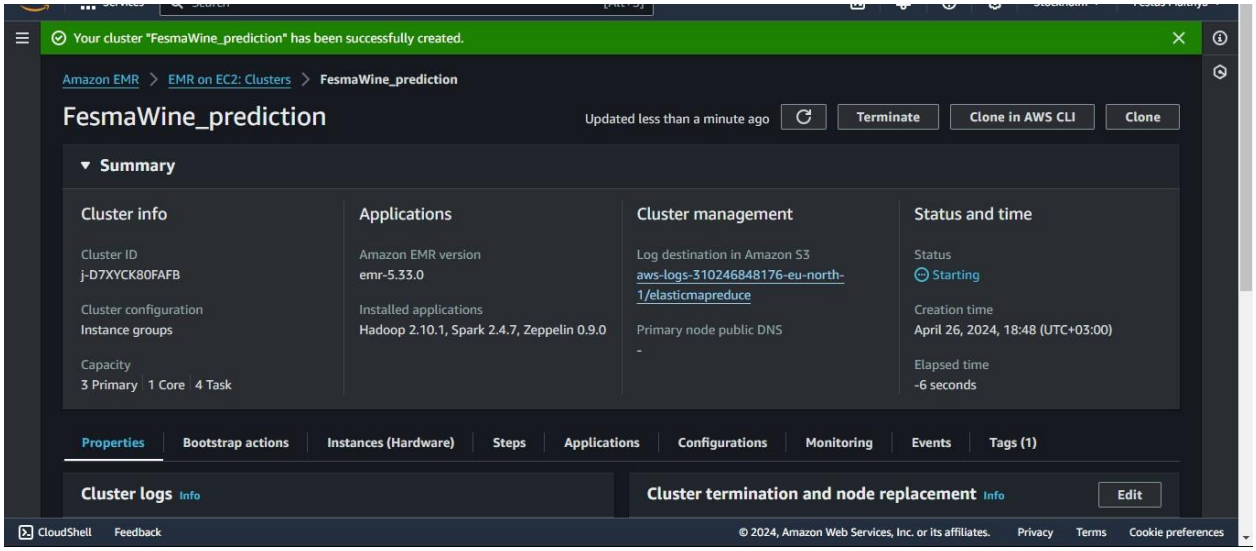
On the Security Configurations, create Key-Pair for EMR cluster by choosing “Create key pair” option on the EC2 key access menu and use .pem as format. This will download {keyname}.pem file which should be kept safe because you we will need it to do SSH to EC2 instances. We can upload it to an S3 bucket.



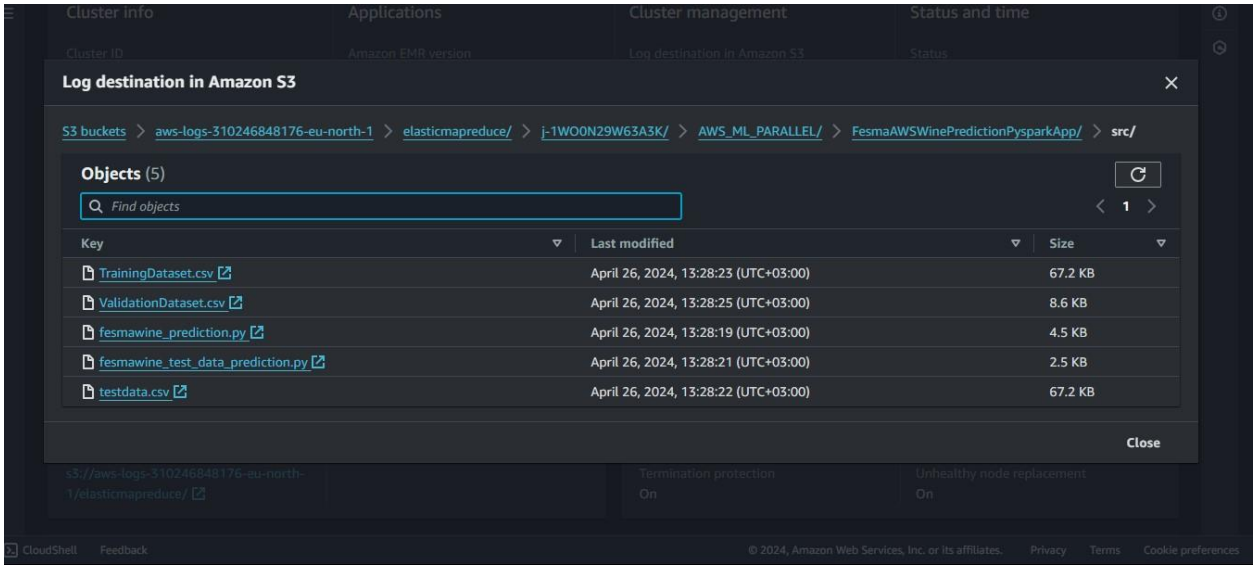


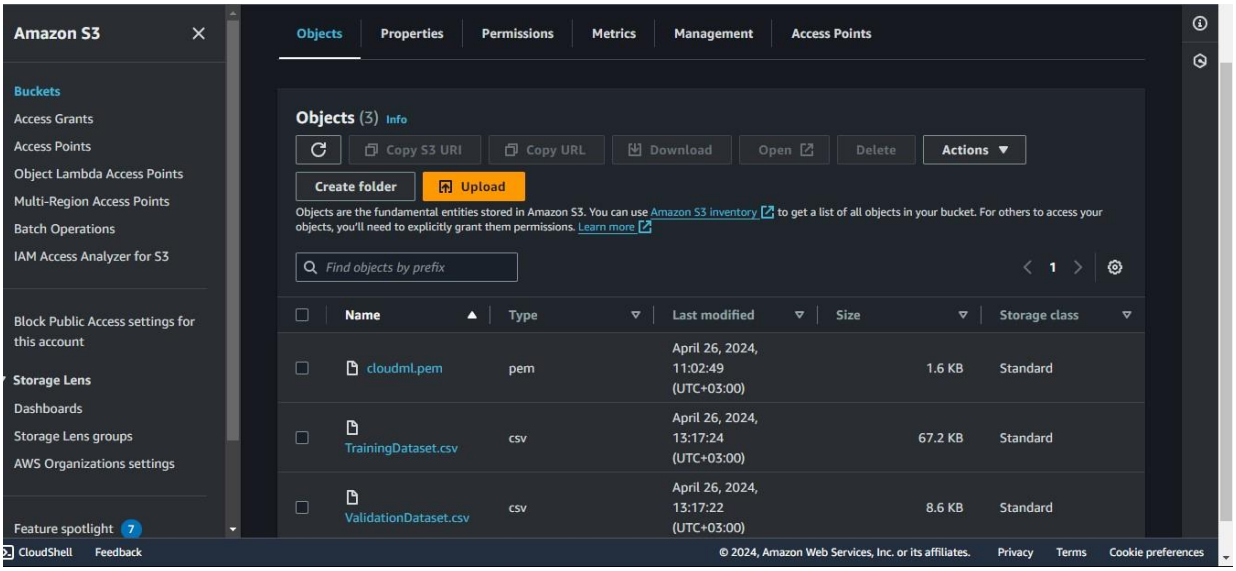


Then we click the “Create Cluster” button to create the EMR cluster.



Here goes a snapshot of the Spark application on EMR.





2:TRAINING ML models in the Spark EMR Cluster in parallel with the 4 EC2 instances.

When the cluster has been provisioned and ready to accept jobs, to submit one you can either use step button to add steps or submit them manually. Doing so manually, we will have to perform SSH to the Master of the cluster using the following cmd:

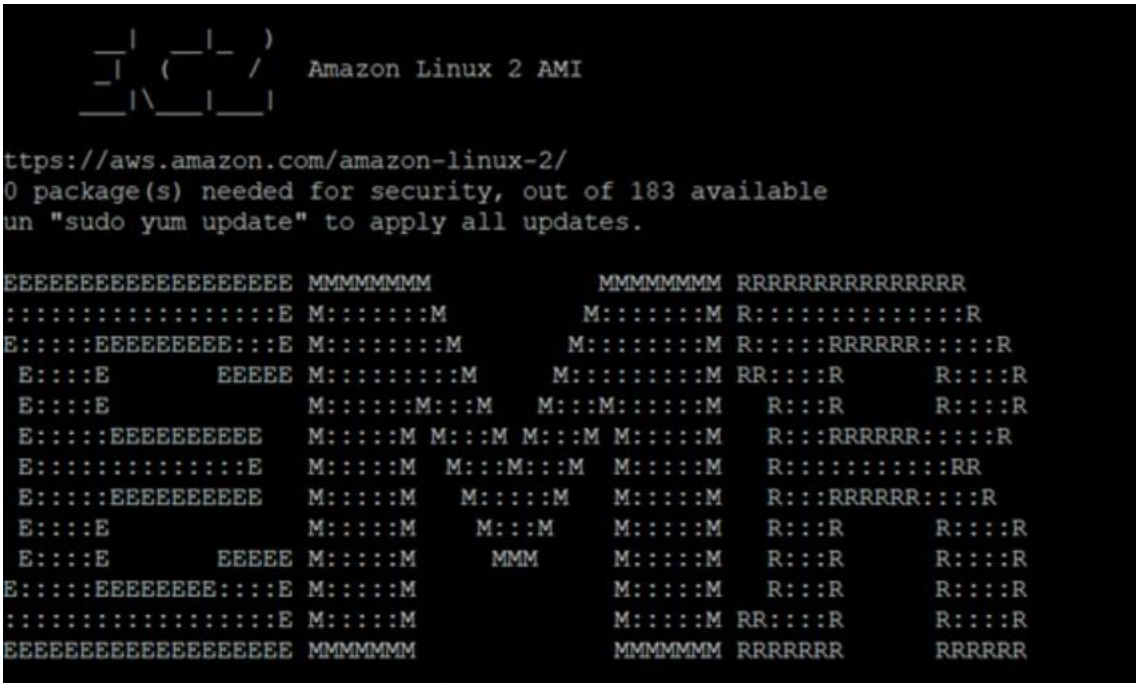
ssh -i "cloudml.pem" <<User>>@<<Public IPv4 DNS>>

And on successful login to master , we change to root user by running the cmd:

sudo su

Then finally we submit the job using following command :

spark-submit s3://{nameofbucket}/fesmawine_prediction.py



Under Ec2 instances click on instances running then connect then it will establish to the connection.

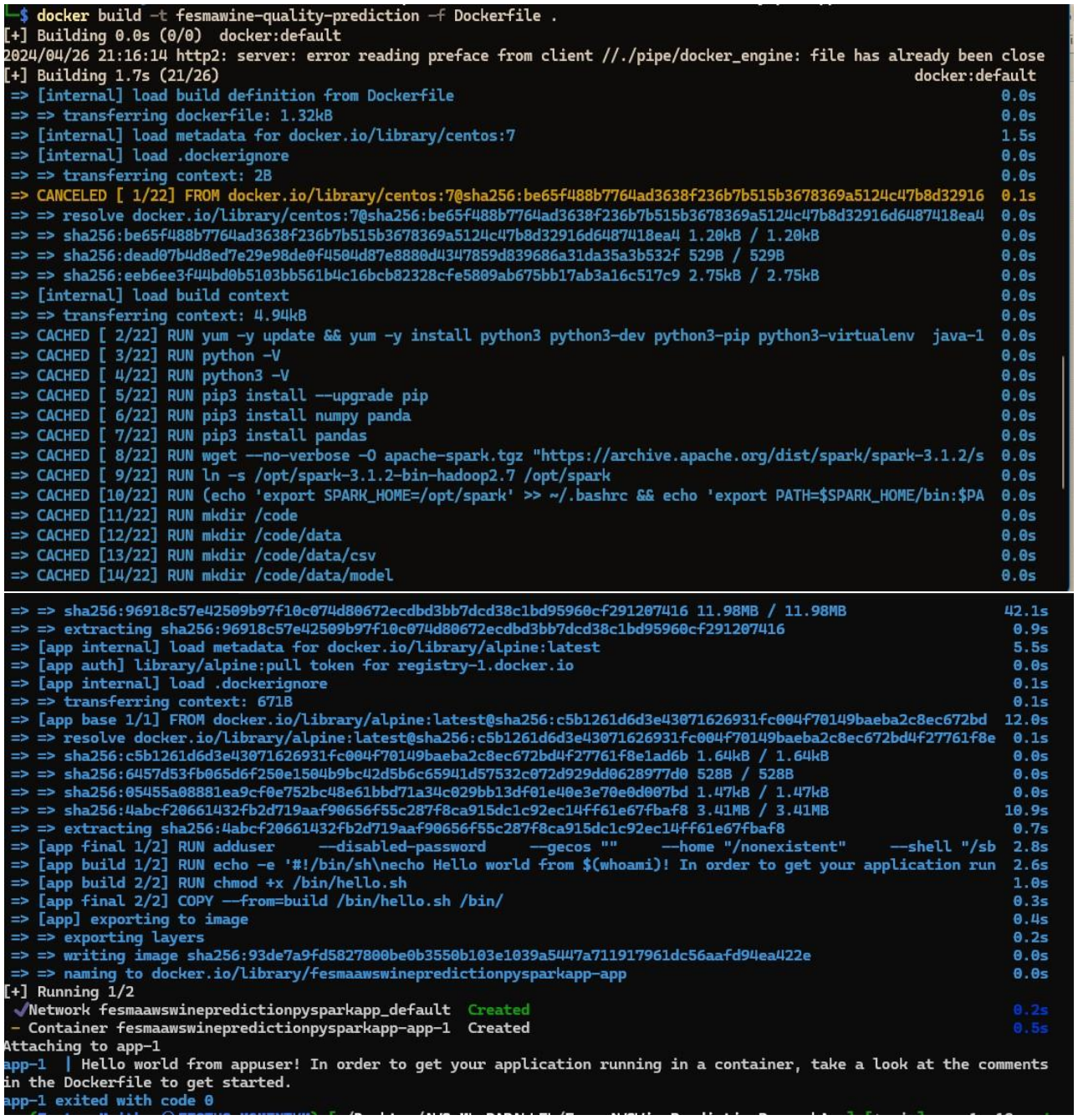
After logging into Emr we have to give permission through Pem key, as we know pem will be generated by using the EMR cluster

We can trace status of the above job in EMR UI application logs. Once status is showing success, a testmodel will be created in s3 bucket-s3://{name of our bucket}.

3.Run ML model using Docker

- **Install docker**
- Build the image of the dockerfile using cmd by typing - **docker build -t fesmawine-qualityprediction**
- You can see the image using cmd by typing - **docker image ls**

- You can push this in docker hub repository by typing - **docker push hemanjali693/fesmaawssparkwinepredictionapp: fesmawine-quality-prediction**



```
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker push hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality
-prediction
The push refers to repository [docker.io/hemanjali693/fesmaawssparkwinepredictionapp]
3212334c57d0: Pushed
5f70bf18a086: Pushed
54a537c48445: Pushed
47e31a4d606a: Mounted from library/python
bf4966b4b813: Mounted from library/python
da15a2a37253: Mounted from library/python
89ca33c95b2e: Mounted from library/python
83db175c22e2: Mounted from library/python
c5d13b2949a2: Mounted from library/python
7e43f593c900: Mounted from library/python
072686bcd3db: Mounted from library/python
fesmawine-quality-prediction: digest: sha256:ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687a1fefc size: 2637
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL>
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker image ls
REPOSITORY                                          TAG                       IMAGE ID       CREATED             SIZE
hemanjali693/fesmaawssparkwinepredictionapp         fesmawine-quality-prediction   32682071a7bc   About an hour ago   2GB
fesmawine-qualityprediction                         latest                    32682071a7bc   About an hour ago   2GB
hemanjali693                                        latest                    32682071a7bc   About an hour ago   2GB
docker/welcome-to-docker                            latest                    c1f619b6477e   5 months ago        18.6MB
hello-world                                         latest                    d2c94e258dcb   12 months ago       13.3kB
```

A public image has been created and posted on DockerHub. Use the command :

docker pull hemanjali693/fesmaawssparkwinepredictionapp: fesmawine-quality-prediction to get the image on your machine.

```
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker pull hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality
-prediction
fesmawine-quality-prediction: Pulling from hemanjali693/fesmaawssparkwinepredictionapp
Digest: sha256:ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687a1fefc
Status: Image is up to date for hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality-prediction
docker.io/hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality-prediction
```

Place your testdata file in a folder (lets call it directory dirA) , which you will mount with docker container and run it using below cmd

docker run -v C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL\dirA fesmawine-qualityprediction testdata.csv

```
----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|      rawPrediction|         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|[8.9,0.22,0.48,0....|  1.0|[82.20804793354075...|[0.54805365290271...|       0.0|
|          7.6|          0.39|      0.31|          2.3|    0.082|               23.0|               71.0| 0.9982|3.
52|     0.65|    9.7|    5.0|[7.6,0.39,0.31,0....|  0.0|[68.39442604639 05...|[0.45596284030927...|       1.0|
|          7.9|          0.43|      0.21|          1.6|    0.106|               10.0|               37.0| 0.9966|3.
17|     0.91|    9.5|    5.0|[7.9,0.43,0.21,0....|  0.0|[95.15405609212 19...|[0.63436037394747...|       0.0|
|          8.5|          0.49|      0.11|          2.3|    0.084|                9.0|               67.0| 0.9968|3.
17|     0.53|    9.4|    5.0|[8.5,0.49,0.11,0....|  0.0|[120.9776033676 84...|[0.80651735578456...|       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|[6.9,0.4,0.14,0.0...|  1.0|[52.9461488794193...|[0.35297432586279...|       1.0|
+-------------+--------------+----------+-------------+---------+-------------------+-------------------+-------+--
--+---------+-------+-------+-------------------+-----+-------------------+--------------------+----------+
only showing top 5 rows

None
Test Accuracy =  0.7935887412040656
Weighted f1 score =  0.772186634151117
```