**CS643 – Programming Assignment 2**

**Steps:**

**GitHub link :** [**https://github.com/rakiiii01/Rakesh-.git**](https://github.com/rakiiii01/Rakesh-.git)

**Docker Repository: https://hub.docker.com/r/rakii23/wine-quality**

# EC2 Instance Step up

1. Create a single Linux or Ubuntu AMI-based EC2 instance using the t2.large image.

2. Now access your ec2 instance using Putty.

3. Update your credentials nano.aws/credentials using vi or nano.

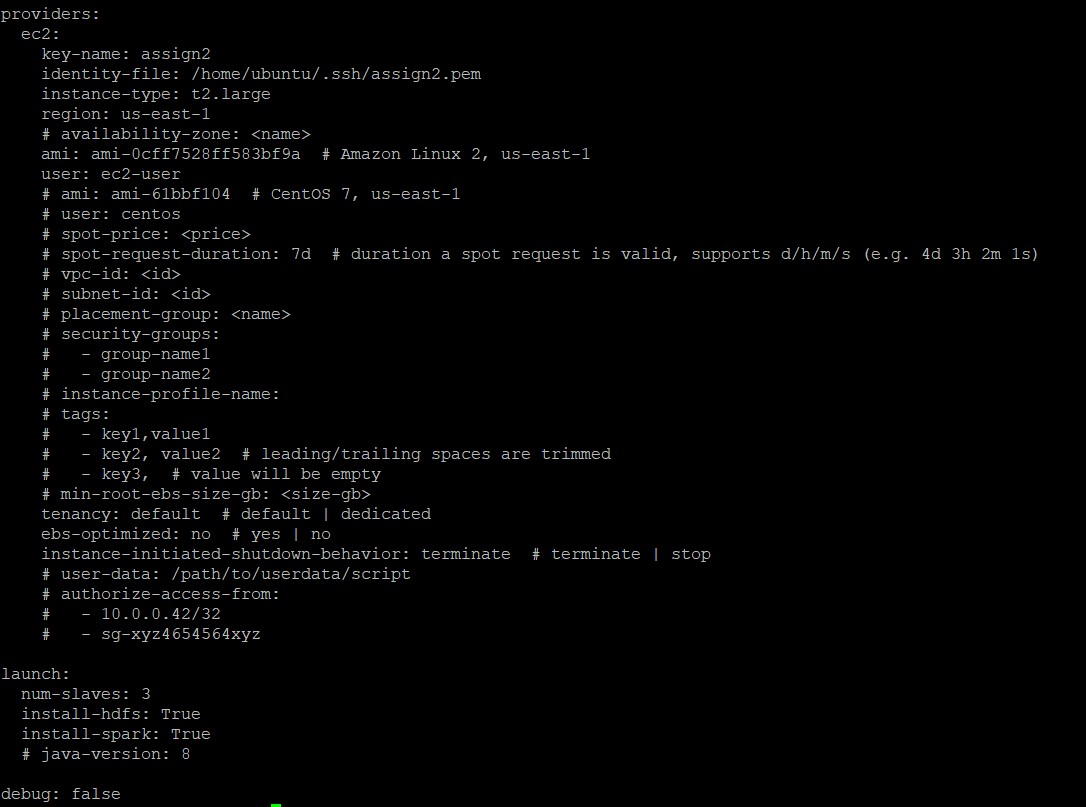
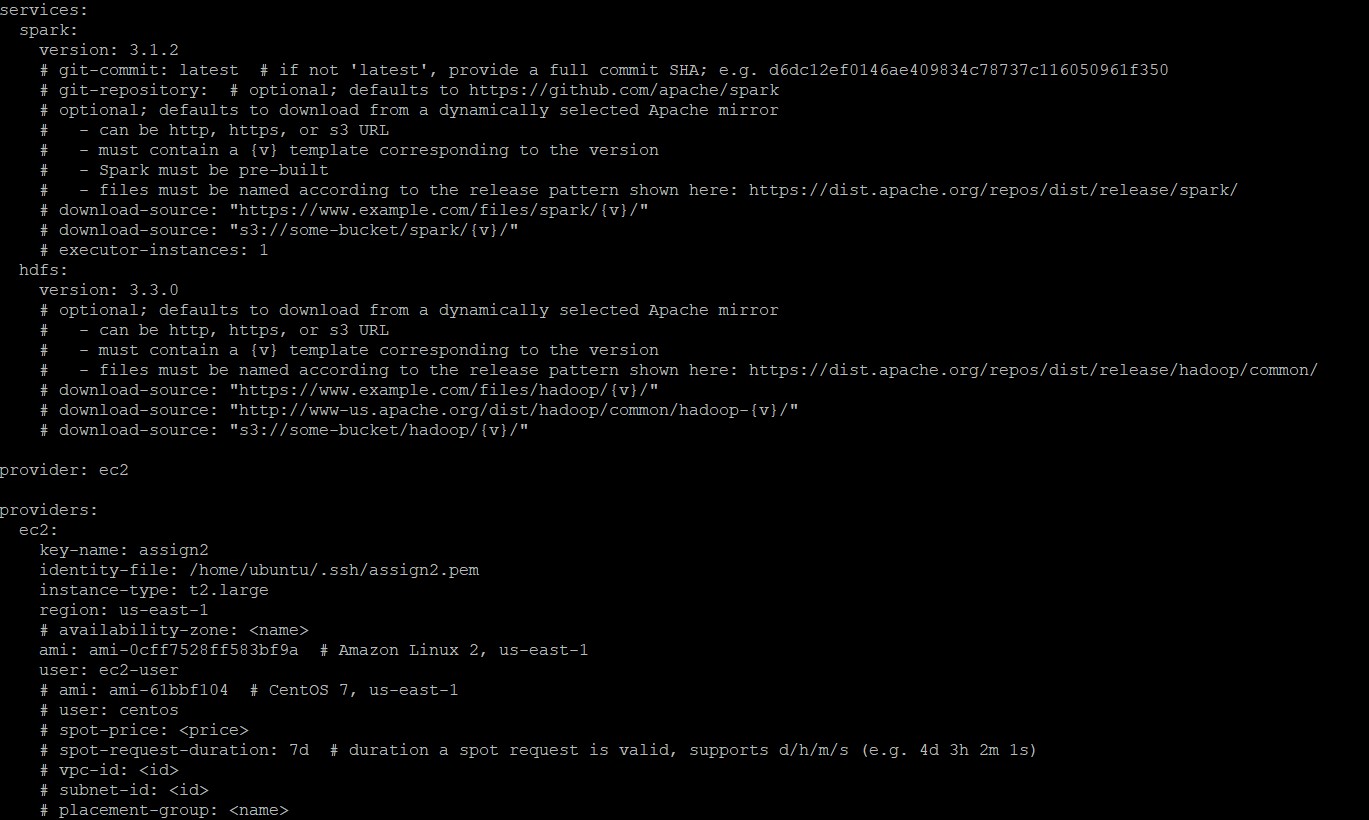
4. Copy the .pem file from your local system to ec2-instance by using SCP command

# Configure Flintrock

1. Install Python and pip on the ec2 instance using the link 2. Now install Flintrock using *pip install flintrock*

3. use ***flintrock configure*** command. This will give you the file location of config.yaml. So go to that location and make changes in the config file using the vi or nano command.

change pem file and file location, use cluster type to t2.large. make sure you install both spark and Hadoop by setting the value True.



4.For launching the cluster use the ***flintrock launch Cluster-name*** command.

5. Finally, log in the cluster using ***flintrock login Cluster-name*** command. You are in your master node now.

Copy Data into the Master node of the cluster

To use SCP command we have to chnage the inbound rules for the master cluster

To do so click on the master node -> security -> security group of flintrock add ssh with 22 port and

0.0.0.0/0

now you can able to use scp command and copy tain.py, test.py ,TainingDataset.csv and ValidationDataset.csv

mkdir assignment

scp -i //Users/rakeshreddynareddy/Desktop/cc /TrainingDataset.csv ec2user@ec2-54-90-98-17.compute-1.amazonaws.com:~/assignment/

# Parallel Taining Implementation across cluster

1. Create a folder in hadoop file system to make datasets globally available for the nodes in the cluster. use

hadoop fs -mkdir /data/

1. Copy files from master node to hdfs hdfs dfs -put /home/ec2-user/assignment/TrainingDataset.csv /data/ hdfs dfs -put /home/ec2-user/assignment/ValidationDataset.csv /data/ cd assignment
2. To verify the files use

hdfs dfs -ls /data

1. Run the following command to create model and parallel traing implementation across your cluster

spark-submit --master spark://ip-172-31-30-207.ec2.internal:7077 train.py hdfs:///data/TrainingDataset.csv hdfs:///model

here ip-172-31-30-207.ec2.internal is the master node address

1. Then we have to get our model using

hdfs dfs -get /model

Text

Description automatically generated

# Single Machine Wine Prediction

On the master node, use

spark-submit --master local[\*] test.py file:///home/ec2-user/assignment/ValidationDataset.csv file:///home/ec2-user/assignment/model\

This will take only master node into consideration and we can see the Accuracy and F1-score on our local machine.

# Setting up Docker on Master node

1. Install most recent docker engine package:

sudo amazon-linux-extras install docker or sudo yum install docker

1. Start docker using

sudo service docker start 3. Adding ec2-user to docker group sudo usermod -a -G docker ec2-user

1. Exit the flintrock cluster and login again.
2. verify ec2-user

docker info

create docker hub account to see your image

1. Create "Dockerfile" in the assignment folder
2. Build the docker image using

"docker build -t winepred ."

1. Tag the docker image usinf docker tag winepred rakii23/wine-quality:tag
2. Login to the docker using

docker login

1. Pushing the image to Docker hub repository

docker push ni rakii/wine-quality:tag

1. Pull the image from the Docker hub repository

docker pull rakii23/wine-quality:tag

# Use Docker Container for wine Prediction

1. Launch your ec2-instance and then step-up docker using the above steps.

Go to Docker repository : <https://hub.docker.com/r/rakii23/wine>[-quality](https://hub.docker.com/r/nidhi247/wine-quality)

1. Pull the image to Docker hub repository

docker pull rakii23/wine-quality:tag

1. Run the image using :

docker run rakii23/wine-quality:tag driver test.py ValidationDataset.csv model

You will see Accuracy and F1-score

Text

Description automatically generated