Programming Assignment 2

Preface

This manual delineates the methodology for constructing an Apache Spark cluster via Flintrock and employing Spark for the assessment of beverage quality. It involves utilizing an established model for analysis and subsequent deployment using Docker services.

Github link:

https://github.com/prasanthreddy9/Cs643_programming_assignment2_prasanth_reddy

Initial Configuration and Setup

Setting Up Flintrock
Firstly, verify if Python 3 is installed. Subsequently, proceed with the Flintrock installation:

pip3 install git+https://github.com/nchammas/flintrock

Configuration of AWS Environment

Configure your EC2 instance with AWS using the command **aws configure**, integrating credentials from the AWS Lab.

Preparing Flintrock
Prior to cluster initialization, verify the availability of a legitimate .pem file for accessing EC2 instances. Follow these steps: Migrate your .pem file to the EC2 instance, Run **flintrock configure** to generate a .config/flintrock/config.yaml file. Then Adjust the .config/flintrock/config.yaml file to include the .pem file path, key-name, identity-file, ami, and adjust the slave count to 4.

Cluster Initiation
Initiate your cluster using Flintrock:

flintrock launch wine-cluster

Data Transfer to Cluster
Upload the Training data file (TrainingDataset.csv) to the cluster:

flintrock copy-file wine-cluster TrainingDataset.csv /home/ec2-user/

Cluster Access
Securely connect to the master node of the cluster:

flintrock login wine-cluster

Library Installation
Install the required libraries within the cluster:
pip3 install numpy
sudo yum install git
Cloning the Repository
Proceed to clone the repository:
git clone <repo_url>

Training Process

This section is dedicated to executing the training operation on the cluster.

Acquiring the Master Node IP

From the AWS EC2 console, retrieve the Public IPv4 DNS of the master node.

Training Execution on the Cluster

Commence the training on the 4-worker cluster:

spark-submit --master spark://<publicIP>:7077 <train.py>

Deployment of Prediction Model

This section deals with setting up a Docker container for the evaluation of beverage quality using the trained model.

Docker Installation on the Cluster
Follow these steps to install Docker on the cluster:

sudo yum install docker
sudo systemctl restart docker
sudo usermod -aG docker $USER

Container Configuration and Deployment
Formulate the Docker container for the beverage quality evaluation service:

docker build -t wine-test .

Deploy the Docker container on the master node:

docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 wine-test:latest

HTML File Modification

Update your local HTML file to direct evaluation requests to the Docker endpoint:

http://<publicIP>:5000/predict

Establishing Inbound Security Rule

Create an inbound security rule via the AWS EC2 dashboard for the master node to permit traffic on port 5000, thus facilitating external access to the Docker service.

Displaying the F1 Score

Following the submission of the validation CSV through the browser, the Docker service will exhibit the evaluation results and the F1 score on the interface.

Completing these steps signifies the successful configuration of a cluster, execution of beverage quality assessment using Spark, and deployment of outcomes via Docker services.