**Spark-Based Beverage Quality Analysis Framework**

**Table of Contents:**

1. Introduction
2. Configuration and Setup Procedures
3. Execution of Model Training
4. Implementation of Prediction Mechanism

**Introduction**

This document outlines the methodology for establishing a Spark cluster via Flintrock, aimed at analyzing beverage quality. It encompasses the use of an existing model for the analysis and the deployment of these results using Docker.

**Configuration and Setup Procedures**

1. Installation of Flintrock
   1. Ensure Python 3 is in place, then proceed with Flintrock installation:

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

1. AWS Environment Configuration
   1. Configure your EC2 instance with AWS using **aws configure**, using credentials from the AWS Lab.
2. Flintrock Environment Preparation
   1. Before embarking on cluster creation, confirm the availability of a valid .pem file for EC2 access. Steps include:
      1. Transferring your .pem file to the EC2 instance.
      2. Running **flintrock configure** to generate a .config/flintrock/config.yaml file.
      3. Modifying the .config/flintrock/config.yaml to include .pem file path, key-name, identity-file, ami, and adjusting the number of slaves to 4.
3. Cluster Initiation with Flintrock
   1. Commence the cluster using Flintrock:

**flintrock launch spark-cluster**

1. Data Migration to Cluster
   1. Transfer the training dataset (TrainingDataset.csv) to the cluster:

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

1. Accessing the Cluster
   1. Log into the master node of the cluster:

**flintrock login spark-cluster**

1. Installation of Required Libraries
   1. In the cluster, install the necessary libraries:

**pip3 install numpy**

**sudo yum install git -y**

1. Repository Cloning Process
   1. Clone the required repository:

**git clone https://github.com/MeleesaJ/assignment2\_meleesa\_jacob.git**

**Execution of Model Training**

This segment addresses running the training process on the cluster.

1. Retrieval of Master Node IP
   1. Obtain the master node's Public IP from the AWS EC2 console.
2. Training Procedure on the Cluster
   1. Begin the training on the 4-worker cluster:

**spark-submit --master spark://PublicIP:7077 train.py**

**Implementation of Prediction Mechanism**

This section focuses on setting up a Docker container for beverage quality assessment using the trained model.

1. Docker Installation on the Cluster
   1. Adhere to these steps to install Docker:

**sudo yum install docker -y**

**sudo systemctl restart docker**

**sudo usermod -aG docker $USER**

**source ~/.bashrc**

1. Docker Container Setup and Deployment
   1. Pull/Build Image Use any 1 option from this section:
      1. Pull Docker image from Hub:

**docker pull meleesaj/wine-flask-image:latest**

* + 1. Build the Docker container for the beverage quality assessment service:

**docker build -t meleesaj/wine-flask-image .**

* 1. Execute the Docker container on the master node:

**docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 meleesaj/wine-flask-image:latest**

1. Local HTML File Adjustment
   1. Modify your HTML file to channel assessment requests to the Docker endpoint: [http://[Public-IPv4-DNS]:5000/predict](about:blank)
2. Creation of Inbound Security Rule
   1. On the AWS EC2 console, establish an inbound security rule for the master node to allow port 5000 traffic, enabling external Docker service access.
3. F1 Score Visualization
   1. After the validation CSV is submitted via a browser, the Docker service will exhibit the assessment results and the F1 score on the user interface.

By following these steps, you'll have successfully established a Spark-based cluster, performed beverage quality analysis, and deployed the analysis results through a Docker service.