**Beverage Quality Assessment Using Apache Spark and Flintrock**

**Table of Contents:**

* Introduction
* System Configuration and Installation
* Conducting the Training Process
* Setting Up the Prediction Interface

**Introduction**

* This guide presents a structured approach to constructing a Spark cluster utilizing Flintrock, focusing on the analysis of beverage quality. The process involves leveraging a pre-built model for analytical purposes and deploying the findings via Docker.

**System Configuration and Installation**

A. Flintrock Installation

Begin by confirming the installation of Python 3. Then, proceed with the following commands for Flintrock installation:

**sudo yum install python3-pip -y**

**sudo yum install git -y**

**pip3 install git+https://github.com/nchammas/flintrock.git@7dde87526c71e2781a2a027d1084dfebe39998c2**

B. AWS Environment Setup

Configure your EC2 instance with AWS credentials using aws configure, based on the AWS Lab guidelines.

C. Preparing Flintrock Environment

Before creating the cluster, ensure you have a valid .pem file for accessing EC2 instances. The steps include:

* Transfer the .pem file to your EC2 instance.
* Run flintrock configure to produce a .config/flintrock/config.yaml file.
* Edit the .config/flintrock/config.yaml file to include the .pem file path, key-name, identity-file, ami, and set the slave count to 4.

D. Initiating Cluster with Flintrock

Start your cluster using Flintrock:

**flintrock launch spark-cluster**

E. Transferring Data to the Cluster

Move the training data (TrainingDataset.csv) to the cluster:

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

F. Cluster Access

Log into the master node:

**flintrock login spark-cluster**

G. Installing Necessary Libraries

Within the cluster, install required libraries:

**pip3 install numpy**

**sudo yum install git**

H. Cloning the Repository

Clone the specified repository:

**git clone** [**https://github.com/PrahasithTR/CC\_PA2.git**](https://github.com/PrahasithTR/CC_PA2.git)

**cd assignment2**

**Conducting the Training Process**

A. Master Node IP Retrieval

From the AWS EC2 console, obtain the Public IP of the master node.

B. Training on the Cluster

Commence the training on the 4-worker cluster:

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

**Setting Up the Prediction Interface**

A. Docker Installation on Cluster

Follow these steps for Docker installation:

**sudo yum install docker**

**sudo systemctl restart docker**

**sudo usermod -aG docker $USER**

B. Docker Container Configuration and Deployment

Pull the Docker container for the assessment service:

**docker pull prahasithtr/flaskapp:latest**

Run the container on the master node:

**docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 prahasithtr/flaskapp:latest**

C. Local HTML File Modification

Update your HTML file to redirect requests to the Docker endpoint: http://[**PublicIP**]:5000/predict

D. Inbound Security Rule Creation

Set up an inbound security rule on the AWS EC2 console for the master node to allow traffic on port 5000.

E. F1 Score Display

Upon submitting the validation CSV in the browser, the Docker service will showcase the assessment results and the F1 score.

