**Initiating a Spark Cluster Configuration**

1. **Flintrock Installation:** Initiate the setup by installing Flintrock, an essential utility for managing Apache Spark clusters. Use the command for installation via pip3.

pip3 install flintrock

1. **Flintrock Configuration:** Post-installation, it is imperative to configure Flintrock by command **flintrock configure** and integrate your EC2 key pair and setting your cluster parameters. The configuration can be updated or created at

.config/flintrock/config.yaml

1. **Cluster Launch:** Deploy the Spark cluster effectively by executing the command:

flintrock launch Dinesh-cluster

,which initiates a cluster with one master and four worker nodes.

1. **Data Transfer to Cluster:** It is crucial to transfer your training dataset, like “TrainingDataset.csv”, to the cluster using the command:

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

1. **Cluster Access:** Access the master node of your Spark cluster by executing:

flintrock login Dinesh-cluster

,and proceed with training operations.

**Training Protocol**

1. **Environment Setup:** Prior to commencing the training process, install Git with the commands:

sudo yum install git

1. **Repository Cloning:** Acquire the training code by cloning the appropriate Git repository using the placeholder:

git clone <repository-url>

cd repo

javac -cp "/home/ec2-user/spark/jars/\*" WineQualityGBTClassification.java

echo Main-Class: WineQualityGBTClassification > Manifest.txt

jar cvfm WineQualityGBTClassification.jar Manifest.txt WineQualityGBTClassification.class

1. **Training Execution:** Conduct the training operation by deploying the training script with:

spark-submit --class WineQualityGBTClassification --master spark://<public-ip>:7077 WineQualityGBTClassification.jar replacing <public-ip> with the IP of your master node.

**Inference Procedure**

1. **Docker Installation:** Install Docker on the cluster with a series of commands, including

sudo yum install docker

sudo systemctl restart docker

and adjusting user permissions with

sudo usermod -aG docker $USER

1. **Container Deployment for Inference:** Within the inference directory, retrieve and initiate the Docker image using:

docker pull dinesh9789/winetest:latest

and docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 dinesh9789/winetest:latest

1. **Interface Setup for Inference:** Modify the HTML file used for inference to link to

http://<public-ip>:5000/predict, ensuring the <public-ip>

