**Establishing a Spark Cluster**

1. **Flintrock Installation Process:** Initiate the cluster setup by installing Flintrock, a specialized tool for managing Apache Spark clusters. This can be installed using the pip3 command:

pip3 install flintrock

1. **Setting up Flintrock Configuration:** Post-installation, it's necessary to configure Flintrock to integrate your AWS EC2 key pair and define cluster settings. You can update or create a new configuration file located at

flintrock configure

.config/flintrock/config.yaml

1. **Cluster Deployment:** For deploying a multi-node Spark cluster, use the command

flintrock launch spark-cluster

,which sets up one master and four worker nodes.

1. **Dataset Integration to Cluster:** To proceed, transfer the training dataset, for instance, TrainingDataset.csv, to the cluster using the command:

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

1. **Master Node Access:** Access the cluster by logging into the master node using the command:

flintrock login spark-cluster

, and then move on to training operations.

**Conducting Training**

1. **Pre-training Setup:** Ensure the necessary tools, such as Git, are installed on the cluster. Execute

sudo yum install git (for Git)

1. **Code Acquisition via Git:** Download the training code by cloning the relevant Git repository. Replace <repository-url> with the actual URL of your Git repository in the command:

git clone <repository-url>

cd repo\_name

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

echo Main-Class: WineQualityClassification > Manifest.txt

jar cvfm WineQualityClassification.jar Manifest.txt WineQualityClassification.class

1. **Training Execution on Cluster:** Execute the training script across the cluster using:

spark-submit --class WineQualityClassification --master spark://<public-ip>:7077 WineQualityClassification.jar

substituting <public-ip> with the public IP of your master node.

**Inference Setup**

1. **Docker Installation and Configuration:** Begin by installing Docker on the cluster with

sudo yum install docker

sudo systemctl restart docker

sudo usermod -aG docker $USER

,then configure system services and user permissions accordingly.

1. **Docker Image Deployment:** Navigate to the directory containing inference code and execute

docker pull rama0322/cloudpa2:latest

Run the Docker container with the command:

docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 rama0322/cloudpa2:latest

1. **HTML File Configuration for Inference:** Update the HTML file designed for inference to direct to

http://<public-ip>:5000/predict

,ensuring that <Public-ip> is correctly replaced.

1. **Security Configuration:** Verify that the security group settings for your master node allow traffic through port 5000 to enable external connections to the Docker-hosted Flask application.

This detailed approach ensures a robust setup for both training and inference phases on your Spark cluster.