**Setting Up a Flintrock Cluster**

1. **Install Flintrock:** Begin your setup by installing Flintrock, a tool designed for launching Apache Spark clusters. Install it easily via pip3 using the command:

pip3 install flintrock.

1. **Configure Flintrock:** After installation, configure Flintrock by command **flintrock configure** and set up your AWS EC2 key pair and define your cluster preferences. Update or create the configuration file located at

.config/flintrock/config.yaml

1. **Launch the Cluster:** Deploy your Spark cluster by running:

flintrock launch test-cluster

This command sets up a cluster with one master and four worker nodes.

1. **Copy Dataset to Cluster:** Transfer your training dataset, such as “TrainingDataset.csv”, to the cluster using:

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

1. **Access the Cluster:** Log into the master node of your Spark cluster with:

flintrock login test-cluster

Once connected, you can proceed to the training steps.

**Training**

1. **Prepare Environment:** Before starting the training, install necessary software Git using:

sudo yum install git

1. **Get Training Code:** Download the specific Git repository that holds your training code by replacing

<repository-url> in: git clone <repository-url>

1. **Execute Training:** Run the training script across your cluster nodes using:

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

echo Main-Class: LRTraining > Manifest.txt

jar cvfm LRTraining.jar Manifest.txt LRTraining.class

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

substituting <public-ip> with your master node’s IP.

**Inference**

1. **Set Up Docker:** Install Docker on the cluster with commands like

sudo yum install docker

sudo systemctl restart docker

and add your user to the Docker group using

sudo usermod -aG docker $USER

1. **Run Inference Container:** Inside the inference directory, pull and start the Docker image with:

docker pull akrao99/awssparkwineclodcomputing:latest and docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 akrao99/awssparkwineclodcomputing:latest

1. **Prepare Inference UI:** Edit the HTML file for your inference interface to connect to http://<public-ip>:5000/predict, updating <public-ip> accordingly.
2. **Adjust Security Settings:** Make sure the security group for your master node allows traffic on port 5000 to enable access to the inference application.

By following these steps, your Spark cluster will be ready for both training and inference tasks.