**Cluster Setup and Wine Quality Testing**

**Table of Contents:**

1. Introduction
2. Installation and Configuration
3. Training
4. Inference

**1. Introduction**

In this document, we will go through the steps to set up a spark cluster using Flintrock and then use Spark to perform wine quality prediction. The prediction will be done using a pre-trained model, and we will serve the prediction results using a Docker container.

**2. Installation and Configuration**

**2.1 Install Flintrock**

To begin, ensure you have Python 3 installed. Then, install Flintrock using the following command:

* pip3 install flintrock

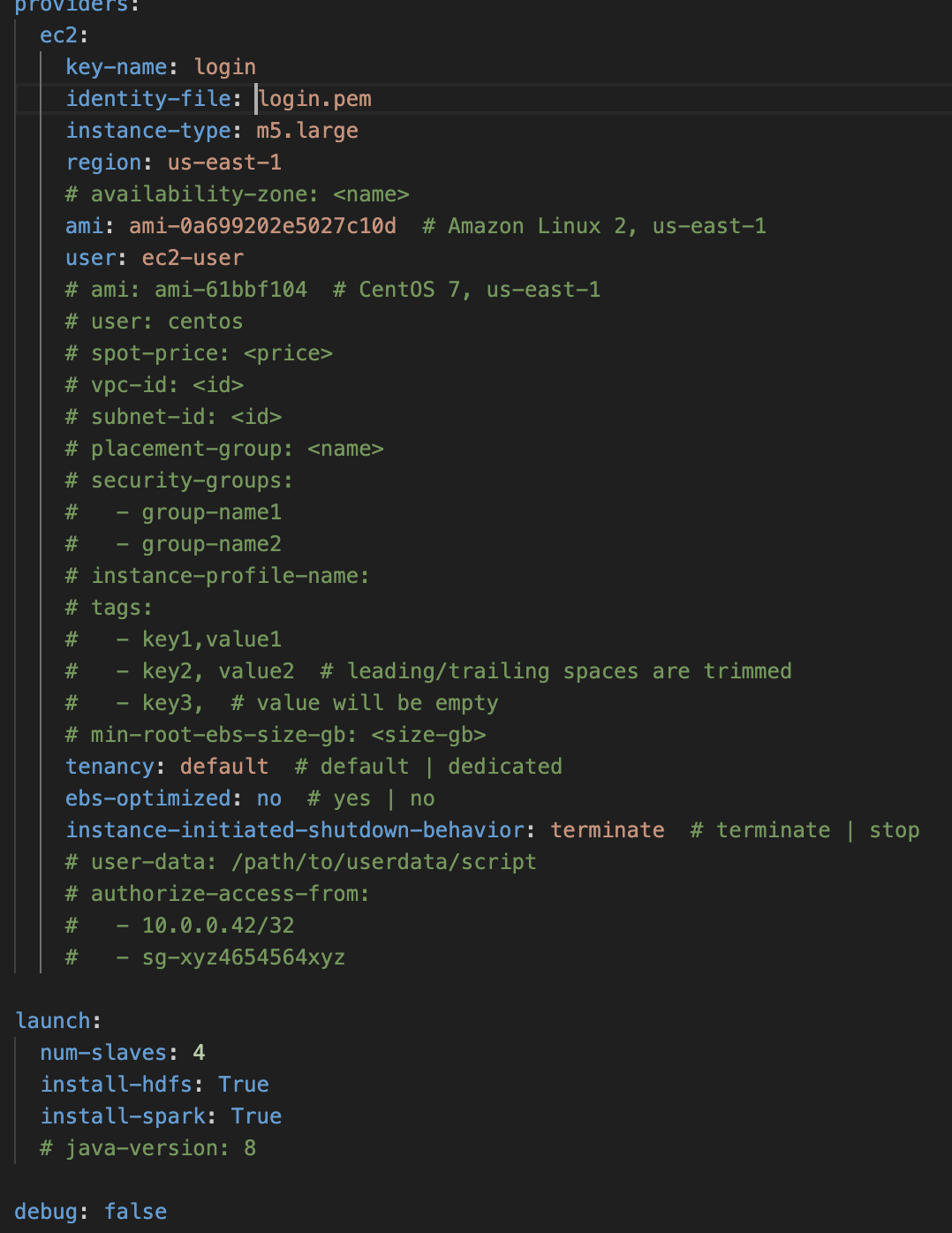
**2.2 Configure AWS**

Configure AWS in the EC2 instance using cmd **aws configure**, and use keys from AWS Lab

**2.3 Configure Flintrock**

Before launching the cluster, make sure you have a valid .pem file to access the EC2 instances. Follow these steps:

* Run flintrock configure to create .config/flintrock/config.yaml
* Update the .config/flintrock/config.yaml file with the path to your .pem file key-name and identity-file, ami, and set the number of desired slaves to 4



**2.3 Create Cluster**

Now, launch the cluster using Flintrock:

* flintrock launch ganesh-cluster

**2.4 Copy Training Dataset to Cluster**

Copy the training dataset (TrainingDataset.csv) into the cluster's master node:

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

**2.5 Login into Cluster**

SSH into the cluster's master node:

* flintrock login ganesh-cluster

**2.6 Install Dependencies**

Inside the cluster, install the required dependencies:

* sudo yum install git

**2.7 Clone Git Code**

Clone the Git repository containing the Spark training code:

* git clone <repository\_url>

**3. Training**

In this section, we will run the training job on the cluster using Spark.

**3.1 Obtain Master Node's Public IP**

From the AWS EC2 dashboard, obtain the public IP address of the master node.

**3.2 Run Training on the Cluster**

Run below commands:

* javac -cp "/home/ec2-user/spark/jars/\*" WineTraining.java
* echo Main-Class: WineTraining > Manifest.txt
* jar cvfm WineTraining.jar Manifest.txt WineTraining.class

Run the training job in the 4-worker cluster using Spark:

* spark-submit --class WineTraining --master spark://[public\_ip]:7077 WineTraining.jar

**4. Inference**

In this section, we will create a Docker container to serve wine quality predictions using the pre-trained model.

**4.1 Install Docker on Cluster**

Install Docker on the cluster by following these steps:

* sudo yum install docker
* sudo systemctl restart docker
* sudo usermod -aG docker $USER

**4.2 Pull and Run Docker Container**

Pull the Docker container for the wine quality prediction service:

* docker pull ganeshkuchana/winequalitytest:latest

Run the Docker container on the master node:

* docker run -v /home/ec2-user/spark:/home/ec2-user/spark -p 5000:5000 ganeshkuchana/winequalitytest:latest

**4.3 Update HTML File**

Update your HTML file to send prediction requests to the Docker container's endpoint: http://[public\_ip]:5000/predict.

**4.4 Display F1 Score on UI**

After submitting the validation CSV on the browser, the Docker container will serve the prediction results along with the F1 score on the user interface.

With these steps completed, you have successfully set up a cluster, performed wine quality prediction using Spark, and served the predictions via a Docker container.

