Setting up the EMR (Can be cloned after initial setup)

1. Use EMR to set up a cluster
   1. emr-7.8.0
   2. Check box for Spark 3.5.4
   3. 5 instances (1 master, 4 slave)
   4. Amazon Linux release 2023.7.20250331.0
   5. EC2 Key Pair: carRekognition
   6. Service Role: EMR\_DefaultRole
   7. Instance Profile: EMR\_EC2\_DefaultRole
2. Add an additional inbound rule to the master node
   1. Type: SSH
   2. Protocol: TCP
   3. Port: 22
   4. Source: My IP

Setting up the S3 (persists after initial setup)

1. Use default configurations
2. Name the S3 “kevin-project-2-files”
3. Add TrainingDataset.csv
4. Add ValidationDataset.csv

Training and Validating the Model (gets saved to S3)

1. Get the master node’s Public IPv4 DNS from the EC2 Dashboard
2. SSH into the master node using MobaXterm
   1. User: hadoop
   2. Private key: select carRekognition.pem
3. Run “sudo python3 -m pip install numpy”
4. Run “nano train\_model.py”
   1. add the train\_model.py script
   2. save it
5. Run “spark-submit train\_model.py”
6. Run “nano predict\_model.py”
7. Run “spark-submit predict\_model.py s3://kevin-project-2-files/ValidationDataset.csv”
8. Check the F1 score

Testing the Model

1. Set up new EC2
2. Set up Docker through MobaXterm SSH
   1. yum update
   2. yum install docker -y
   3. service docker start
   4. usermod -a -G docker ec2-user
   5. docker --version
   6. restart SSH session
3. Add Dockerfile and predict\_model.py to EC2
4. Run “**docker build -t my-spark-app .**”
5. To run with S3 file:
   1. Add **TestDataset.csv** to the S3
   2. Run “**docker run my-spark-app s3://kevin-project-2-files/TestDataset.csv**”
6. To run with file uploaded to EC2:
   1. Add **TestDataset.csv** to the EC2 with the MobaXterm GUI
   2. Run “**docker run -v /home/ec2-user:/data my-spark-app /data/ValidationDataset.csv**”