GIT: https://github.com/vivekbaskar1405/QualityTrainingEngine.git https://github.com/vivekbaskar1405/QualityPredictionEngine.git

Docker Hub: docker pull vivekbaskar92/cloudcomputing:QualityPredictionEngine

a) Steps to Application without Docker Container

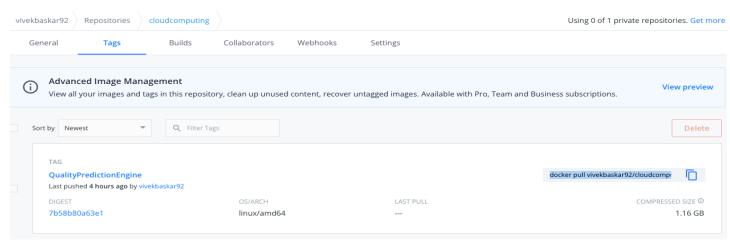
1) Launch Lab, This will launch all the EC2 instance as shown below:

~	spark-cluster-cloud-master	I-0482d7960fef7ca34		t2.micro	⊘ 2/2 checks passed No alarms + us-east-16	ec2-54-198-197-61.co	54.198.197.61
	spark-cluster-cloud-slave	i-07fe07669860215ba	⊘ Running	t2.micro	② 2/2 checks passed No alarms + us-east-16	ec2-174-129-48-118.co	174.129.48.118
	spark-cluster-cloud-slave	I-0c7278a32fc3bdff6	⊗ Running	t2.micro	⊘ 2/2 checks passed No alarms + us-east-16	ec2-54-160-201-73.co	54.160.201.73
	spark-cluster-cloud-slave	i-0b26efe1498aaf0ef	⊗ Running	t2.micro	② 2/2 checks passed No alarms + us-east-16	ec2-54-159-90-56.com	54.159.90.56
	spark-cluster-cloud-slave	i-085566972a01341e1	⊗ Running	t2.micro	⊘ 2/2 checks passed No alarms + us-east-16	ec2-3-90-29-152.comp	3.90.29.152
~	spark-ec2	i-0ffb44572c2bff6b4	⊗ Running	t2.micro	② 2/2 checks passed No alarms + us-east-10	ec2-18-212-233-85.co	18.212.233.85

- 2) SSH into "spark-cluster-cloud-master"
- 3) Run.ksh is created to run the QualityTrainingEngine to create Model and then QualityPredictionEngine to validate the model.
- 4) Run "./run.ksh" to run the application which will Create Training.log and Prediction.log
- 5) Model will be created in /home/ec2-user/model

b) Steps to Run Prediction Application as Docker Container

- 1) SSH into EC2 instance named as "spark-ec2"
- 2) Run "sudo docker pull vivekbaskar92/cloudcomputing:QualityPredictionEngine



- 3) Run "sudo docker images"
- 4) Copy the Files into /home/ec2-user/files

- 5) Rename the data file as ValidationDataset.csv
- 6) cd /home/ec2-user/files (Volume mount)
- 7) Run → sudo docker run -v \$PWD:/opt/bitnami/local/files 'imageld' Eg : "sudo docker run -v \$PWD:/opt/bitnami/local/files 5adb3e8eaf50"
- 8) This will run the docker container in spark-ec2 instance.