**Spark With Kubernetes Setup**

**Chandrashekhar**

Main motive behaind , integration of Spark and kubernetes is to make a centraize cluster where spark jobs is executing and terminating irrespective of project.

There are 3 major parts of the Spark Architecture.

**Driver** — Responsible for breaking up an incoming spark job into smaller junks and govern the execution of those chunks

**Executor** — Responsible for running an individual chunk of the overall job and reporting the result back to the driver.

**Cluster Manager** — Kubernetes infrastructure on which the Spark driver and Executor would run in Pod.Earlier Spark were used with Yarn, Mesos, and Standalone cluster managers but this setup is with kubernetes

**Software required and Version**-

Apache Spark 3.3.0

Docker 1.41

Kubectl v1.25.0

**Rediness of K8 cluster**:

--Used kind 3 node cluster.

Implement or create below accounts before

kubectl create serviceaccount spark

kubectl create clusterrolebinding spark-role --clusterrole=edit --serviceaccount=spark:spark --namespace=default

**Reference Docker image**

cnsnoida/test-spark:3.3.0

**Steps to make Image:**

Download Official Spark from site.

wget <https://archive.apache.org/dist/spark/spark-3.3.0/spark-3.3.0-bin-hadoop2.tgz>

OR

wget https://archive.apache.org/dist/spark/spark-3.3.0/spark-3.3.0-bin-hadoop3.tgz

Extract tgz File and inside spark folder below files will require to check/create. Either change in file or replace with Attached file. If you want to use tini package check the official documentation. Although we have not used tini package.

SPARK\_HOME/bin/docker-image-tool.sh

SPARK\_HOME/kubernetes/dockerfiles/spark/Dockerfile

SPARK\_HOME/kubernetes/dockerfiles/spark/entrypoint.sh

SPARK\_HOME/kubernetes/dockerfiles/spark/start-spark.sh

**docker-image-tool.sh**

No change

**entrypoint.sh (commented tini pkg) see the last 3 lines**

# Execute the container CMD under tini for better hygiene

#exec /usr/bin/tini -s -- "${CMD[@]}"

exec ${CMD[@]}

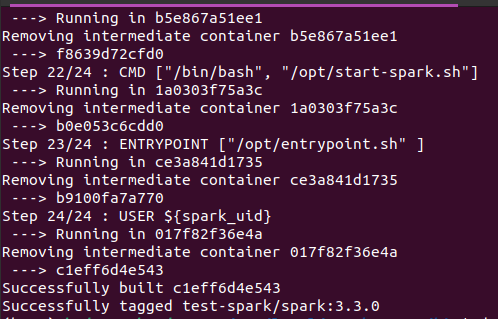
**start-spark.sh**

- create this file or copy and paste in designated folder

**Execution for image. Once build is successful then tag and push to your repository**

./docker-image-tool.sh -r test-spark -t 3.3.0 build





**Test spark jobs:**

1. K8 Yaml

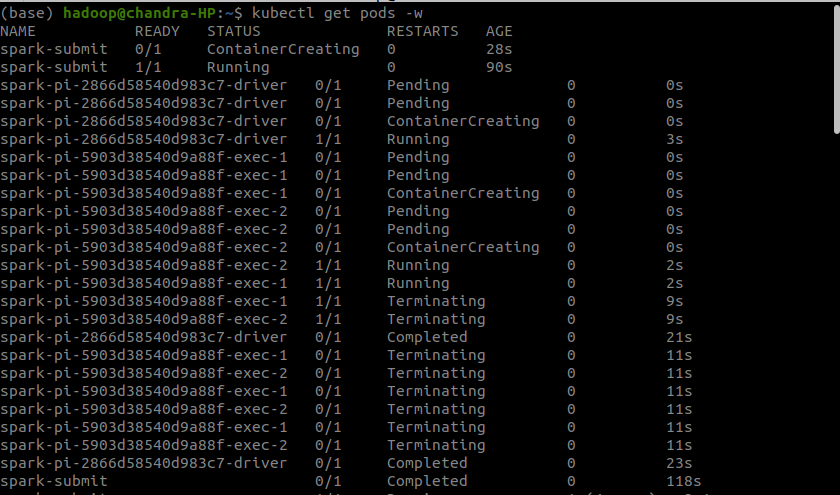
Sample Pod yaml (test.yaml) is attached or create your job accordingly. We have used sample **Pi java program** which is shipped with spark download.

Kubectl apply -f test.yaml #pod

Kubectl apply -f job\_java.yaml #Job

Try to open another cmd console and apply k8 watcher to see live pods creation.

Kubectl get pods -w



2. spark submit CMD

Spark-submit should be in local classpath and jar should be in your local file .Also check cluster API

-kubectl cluster-info

--currently not working

$SPARK\_HOME/bin/spark-submit --master k8s://https://127.0.0.1:44343 --deploy-mode cluster --name spark-pi --class org.apache.spark.examples.SparkPi --conf spark.executor.instances=2 --conf spark.kubernetes.authenticate.driver.serviceAccountName=spark --conf spark.kubernetes.container.image=cnsnoida/test-spark:3.3.0 --conf spark.kubernetes.file.upload.path=/home/hadoop/spark-3.3.1/examples/jars/spark-examples\_2.12-3.3.1.jar

3. Java program

fgdf

4. Operator

Use Helm chart

5. Java API

Spark Submit

http://<ip>:<port>/v1/submissions/create

Spark Kill

http://<ip>:<port>/v1/submissions/kill

Spark Status

http://<ip>:<port>/v1/submissions/status