Reference : https://www.youtube.com/watch?v=1CGGTMvy67c&t=562s

choco install kind ( run cmd as administrator)

-------Using Spark Shell with delta---------

* kind create cluster --name spark
* kubectl config use-context kind-spark
* kind get clusters
* docker build -t spark-kube:3.2.0 .

(required files : <https://github.com/testdrivenio/spark-kubernetes/tree/master/docker>)

(note : dont tag it as latest , it will fail in kind )

* kind load docker-image spark-kube:3.2.0 --name spark

( docker image is loaded into spark-control-plan )

* docker exec -it spark-control-plane crictl images
* ( then deploy master , worker through yaml )
* kubectl get pods -o wide -n default

( note the name and ip of master )

* kubectl exec -n default spark-master-75897b665c-ccczf -it -- pyspark --conf spark.driver.bindAddress=172.17.0.8 --conf spark.driver.host=172.17.0.8

running python files:

kubectl config use-context kind-demo

kubectl get pods -o wide -n default

kubectl cp ./ spark-master-86d5795cff-wdd5m:/demo

(Copy local files into master )

kubectl exec -n default spark-master-86d5795cff-wdd5m -it bin/bash

spark-submit --conf spark.driver.bindAddress=172.17.0.5 --conf spark.driver.host=172.17.0.5 --total-executor-cores 1 --executor-memory 512m sample.py

spark-submit --conf spark.driver.bindAddress=172.17.0.5 --conf spark.driver.host=172.17.0.5 --total-executor-cores 1 --executor-memory 512m --packages io.delta:delta-core\_2.12:1.2.1 ./demo/df.py

spark-submit --conf spark.driver.bindAddress=172.17.0.5 --conf spark.driver.host=172.17.0.5 --total-executor-cores 1 --executor-memory 512m --driver-class-path ./opt/spark/jars/postgresql-42.2.5.jar ./demo/post.py

readlink -f

( export SPARK\_EXECUTOR\_MEMORY=2g

minikube --memory 8192 --cpus 2 start )

done!!!

( Next steps are not essential now )

------for production using operator--------

kind create cluster --name production

kubectl config use-context kind-production

kubectl apply -f spark-rbac.yml -n default

helm repo add spark-operator https://googlecloudplatform.github.io/spark-on-k8s-operator

docker pull gcr.io/spark-operator/spark-operator:3.1.1

kind load docker-image gcr.io/spark-operator/spark-operator:3.1.1 --name production

helm install sparkrel spark-operator/spark-operator -f values.yml -n default

start running jobs....

1. kubectl apply -f spark-pi.yml -n default

2. kubectl apply -f spark-pi-py.yml -n default

3.