Nicola Santillo. 24.12.2023

KUBERNETES ENABLEMENTS:

Set enable from docker desktop

To reset click reset from docker desktop in kubernetes

KIND INSTALLATION:

Open as a Administrator (right click on start and choose Terminal (admin))

curl.exe -Lo kind-windows-amd64.exe https://kind.sigs.k8s.io/dl/v0.20.0/kind-windows-amd64

mv .\kind-windows-amd64.exe "C:\Program Files\Docker\Docker\resources\bin\kind.exe"

To connect to a node:

docker exec -it my-node-name bash

Kind Architecture:

If there are multiple master's nodes you need to connect to the load balancer. Then it distributes the request to the masters

All docker containers are on the same network "kind"

INSTRUCTION:

Create a cluster named "llama" using specified configuration file "clstr-config.yml"

kind create cluster --name llama --config clstr-config.yml

Export the kubeconfig file for a specified cluster

kind export kubeconfig --name llama

Build a Docker image for the application and name it : llama-webapp-img.

docker build . -t llama-webapp-img

Apply the configuration file to the cluster. Patch kind to forward the hostPorts to an NGINX ingress controller and schedule it to the control-plane custom labelled node

kubectl apply -f ingress-deploy.yml

Waiting NGINX ingress controller is ready

kubectl wait --namespace ingress-nginx --for=condition=ready pod --selector=app.kubernetes.io/component=controller --timeout=180s

Load the Docker image into the Kind cluster

kind load docker-image llama-webapp-img --name llama

Deploy a pod named "llama-webapp1" on node "llama-worker”

```
kubectl apply -f pod1.yml

# Deploy a pod named "llama-webapp2" on node "llama-worker”

kubectl apply -f pod2.yml

# kubectl get pods -o wide

# Create services for the pods, exposing port 7860

kubectl apply -f service1.yml

kubectl apply -f service2.yml

#kubectl get svc

#kubectl get pods

# kubectl get nodes

# Create a load balancer service to route traffic to one or other pod

kubectl apply -f loadbl-service.yml

kubectl apply -f nginx-ingress.yml

#kubectl describe ingress nginx-ingress

#  Apply the MetalLB manifest

kubectl apply -f metallb-native.yml

kubectl wait --namespace metallb-system --for=condition=ready pod --selector=app=metallb --timeout=90s

# Configure metallb to use an IP range from the network Docker

docker network inspect -f '{{.IPAM.Config}}' kind

# Assign to our loadbalancer an external IP address

kubectl apply -f metallb-config.yml

# kubectl get pods -n metallb-system --watch

kubectl apply -f nginx-ingress.yml
```

USEFUL INSTRUCTIONS:

```
# docker exec -it llama-worker crictl images / docker exec -it llama-control-plane crictl images

kubectl port-forward svc/llama-webapp 7860:7860

# kind delete cluster --name llama

 # rm C:\Users\nicol\.kube\config

#kubectl uncordon llama-worker

# kubectl delete pod llama-webapp2
```

Connect to the app from another host on port 80:

Ipconfig to retrieve the ip address of the host (Indirizzo IPv4. . . . . : 192.168.1.175)

References:

https://docs.docker.com/get-started/kube-deploy/

https://joe.blog.freemansoft.com/2020/07/multi-node-kubernetes-with-kind-and.html

https://kind.sigs.k8s.io/docs/user/quick-start/#creating-a-cluster

https://kind.sigs.k8s.io/docs/user/quick-start/#configuring-your-kind-cluster

https://www.youtube.com/watch?v=kkW7LNCsK74 (17.26)

https://stackoverflow.com/questions/62694361/how-to-reference-a-local-volume-in-kind-kubernetes-in-docker

https://kk-shichao.medium.com/expose-service-using-nginx-ingress-in-kind-cluster-from-wsl2-14492e153e99

https://medium.com/groupon-eng/loadbalancer-services-using-kubernetes-in-docker-kind-694b4207575d