Openfaas installation steps on Minikube

- Update System:

  sudo apt-get update

  sudo apt-get install apt-transport-https

  sudo apt-get upgrade

- Download minikube:

  wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

  chmod +x minikube-linux-amd64

  sudo mv minikube-linux-amd64 /usr/local/bin/minikube

  minikube version

- Install kubectl:

  curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

  chmod +x ./kubectl

  sudo mv ./kubectl /usr/local/bin/kubectl

  kubectl version -o json

- Starting minikube:

  minikube start

- Check Nodes:

  kubectl get nodes

- Faas-cli

  curl -sL cli.openfaas.com | sudo sh

- Install Helm

  curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

  chmod 700 get_helm.sh

  ./get_helm.sh

  helm version

- Create openfaas namespace in Minikube:

  kubectl apply -f https://raw.githubusercontent.com/openfaas/faas-netes/master/namespaces.yml

- Add openfaas helm repository:

  helm repo add openfaas https://openfaas.github.io/faas-netes/

- Update all helm charts:

  helm repo update

- Generate a random password:

```
export PASSWORD=$(head -c 12 /dev/urandom | shasum| cut -d' ' -f1)
```

- Display and note the password:

```
echo $PASSWORD
```

- Creation of basic-auth for openfaas:

```
kubectl -n openfaas create secret generic basic-auth --from-literal=basic-auth-user=admin --from-literal=basic-auth-password="$PASSWORD"
```

- Install openfaas

```
helm upgrade openfaas --install openfaas/openfaas --namespace openfaas --set functionNamespace=openfaas-fn --set basic_auth=true
```

- Set openfaas_url as an env-var:

```
export OPENFAAS_URL=$(minikube ip):31112
echo $OPENFAAS_URL
kubectl get pods
```

- Display all the namespaces:

```
kubectl get namespaces
kubectl config current-context
```

- Change namespace:

```
kubectl config set-context minikube --namespace openfaas
```

- Get containers status

```
kubectl get pods
```

```
echo -n $PASSWORD | faas-cli login -g http://$OPENFAAS_URL -u admin — password-stdin
```

- Life-cycle of functions:

1. Create:

```
faas-cli new --lang python3 hello
```

2. Build:

```
faas-cli build -f hello.yml
```

```
docker login
```

3. Push:

```
faas-cli push -f hello.yml
```

4. Deploy:

```
faas-cli deploy -f hello.yml --gateway http://$(minikube ip):31112
```

OR, execute 'faas-cli up' command to build, push and deploy a function to openfaas in a single run

```
faas-cli up -f hello.yml --gateway http://$(minikube ip):31112
```

5. Invoke:

   a. Through CLI

   ```
   echo test | faas-cli invoke hello --gateway http://$(minikube ip):31112
   ```

   b. Through openfaas-ui

- List all functions:

  ```
  faas-cli list --gateway http://$(minikube ip):31112
  ```

- Monitor the functions:

  ```
  kubectl get deployments
  ```

  a. Expose deployment:

     ```
     kubectl expose deployment prometheus --type=NodePort --name=prometheus-ui
     ```

  b. View the prometheus-ui service

     ```
     kubectl get svc prometheus-ui
     ```

  c. Open prometheus-ui on local system

     ```
     kubectl port-forward svc/prometheus-ui 9090:9090 &
     ```

- Visualize:

  a. Create Grafana pod:

     ```
     kubectl run grafana --image=stefanprodan/faas-grafana:4.6.3 --port=3000
     ```

  b. Expose the pod:

     ```
     kubectl expose pod grafana --type=NodePort --name=grafana
     ```

  c. View the Grafana service

     ```
     kubectl get service Grafana
     ```

  d. Open Grafana on local system

     ```
     kubectl port-forward svc/grafana 3000:3000 &
     ```

- References:

  https://medium.com/faun/getting-started-with-openfaas-on-minikube-634502c7acdf