

Im Dripping

(real dripping)

[sorry ga terlalu jago, nikmati dokumentasi ampas saya]

Requirements

Pertama-tama saya mau memberi tahu terlebih dahulu jikalau saya mengerjakan soal ini pada Windows. Sehingga segala hal yang mungkin berbeda pada macOS atau Linux (saya angkat tangan). Ok let's go, ada beberapa hal yang dibutuhkan untuk mengerjakan soal ini.

- Source code web aplikasi yang akan di-deploy
- Docker
- Kubernetes
- Minikube (saya make minikube)

***Note:** ini saya pake minikube, tapi tetep di-local kalau di-cloud saya bingung karena udah cukup mabok sama pelabuhan, tapi pake minikube 😊

Install

1. Docker: <https://docs.docker.com/desktop/install/windows-install/>
2. Kubernetes: <https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>
3. Minikube: <https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download>

Setup

Yak, kita masuk ke bagian setup. Sebelum itu ada beberapa hal lagi yang disiapkan yaitu, script deployment dan service yang nantinya akan digunakan.

Service.yml

```
apiVersion: v1 # versi dari API yang digunakan
kind: Service # jenis resource yang akan dibuat
metadata: # metadata dari resource yang akan dibuat
  name: expense-tracker-service # nama service
spec: # spesifikasi dari service
  type: NodePort # tipe service yang akan dibuat
  selector: # selector untuk menentukan pod yang akan diatur
    oleh service
    app: expense-tracker # label app dengan value expense-
    tracker
  ports: # port yang akan diexpose oleh service
    - protocol: TCP # protokol yang digunakan
      port: 80 # port 80
      targetPort: 80 # port yang dituju
      nodePort: 30000 # port yang akan diexpose ke node
```

Deployment.yml

```
# bagian ini untuk mendefinisikan deployment
apiVersion: apps/v1
# kind untuk mendefinisikan jenis resource yang akan dibuat
kind: Deployment
# metadata untuk mendefinisikan metadata dari resource yang
akan dibuat
metadata:
  name: expense-tracker-deployment # nama deployment
spec: # spesifikasi dari deployment
  replicas: 1 # jumlah replika yang akan dibuat
  selector: # selector untuk menentukan pod yang akan diatur
    oleh deployment
    matchLabels: # label yang akan dijadikan acuan oleh
    selector
    app: expense-tracker # label app dengan value expense-
    tracker
  template: # template untuk mendefinisikan pod yang akan
    dibuat
    metadata: # metadata dari pod yang akan dibuat
      labels: # label dari pod yang akan dibuat
        app: expense-tracker # label app dengan value expense-
        tracker
    spec: # spesifikasi dari pod yang akan dibuat
      containers: # container yang akan dijalankan oleh pod
        - name: expense-tracker # nama container
```

```
image: kiiiro/kubernetes-dripping:latest # image yang akan digunakan
ports: # port yang akan diexpose oleh container
- containerPort: 80 # port 80
```

Ok, lanjut

1. Pertama buat image terlebih dahulu dari webapp yang akan di-deploy. Berikut cuplikan dari dockefile yang ada

Figure 0-1 dockerfile untuk membuat image dari webapp

Selanjutnya, buat sebuah repository pada <https://hub.docker.com> yang nantinya image akan di-push ke docker hub (tbh, saya ga yakin part ini buat apa, karena keknya image yang dipake tetep yang local).

Eits, pastikan terminal ada di posisi yang sama dengan adanya dockerfile.

```
# build image
docker build -t kiiiro/Kubernetes-dripping .

# push docker image ke docker hub
Docker push kiiiro/Kubernetes-dripping
```

Nah, harusnya di local sudah muncul docker image yang telah di-build barusan

```
PS D:\Indra\Programming\seleksi-lab\seleksi-sister\14.I-Am-Dripping> docker images
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE
kiiiro/kubernetes-dripping                latest   f4f89e64633d  23 hours ago  43.8MB
kiiiro/client                             latest   4f1a1718241b  3 days ago   1.4GB
react-expense-tracker-backend-client       latest   95c370115232  3 days ago   1.4GB
react-expense-tracker-backend-backend     latest   b39af8c0d43b  3 days ago   1.13GB
kiiiro/backend                             latest   243a7012df2e  3 days ago   1.13GB
mongo                                      latest   a31b196b207d  7 weeks ago   796MB
gcr.io/k8s-minikube/kicbase                v0.0.44  5a6e59a9bdc0  3 months ago  1.26GB
PS D:\Indra\Programming\seleksi-lab\seleksi-sister\14.I-Am-Dripping>
```

Figure 0-2 Hasil Docker Image

2. Langkah selanjutnya adalah pembuatan cluster pada Kubernetes

1. Jalankan command

- minikube start -p <nama_profile>

```
PS C:\Users\DELL> minikube start -p dripping
(minikube) minikube v1.33.1 on Microsoft Windows 11 Home Single Language 10.0.22631.4037 Build 22631.4037
* Automatically selected the docker driver. Other choices: virtualbox, ssh
* Using Docker Desktop driver with root privileges
* Starting "dripping" primary control-plane node in "dripping" cluster
* Pulling base image v0.0.44 ...
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "dripping" cluster and "default" namespace by default
```

Oke, ini berhasil dibuat

2. Cek, apakah kubectl menggunakan context yang sudah anda buat atau belum, dengan menggunakan command

- kubectl config current-context

```
PS C:\Users\DELL> kubectl config current-context
dripping
```

Kalo semisal, berbeda bisa gunakan perintah ini

- kubectl config use-context <nama_cluster>

contoh kasus:

```
PS C:\Users\DELL> kubectl config use-context minikube
Switched to context "minikube".
PS C:\Users\DELL> kubectl config current-context
minikube
PS C:\Users\DELL> kubectl config use-context dripping
Switched to context "dripping".
```

3. Nah, karena kubectl sudah menggunakan cluster yang sesuai kita bisa apply script service.yml dan deployment.yml pada cluster “dripping”

Gunakan command:

- kubectl apply -f <script_deploy_klen.yml>
- kubectl apply -f <script_service.klen.yml>

```
PS D:\Indra\Programming\seleksi-lab\seleksi-sister\14.I-Am-Dripping> kubectl apply -f deployment.yml
deployment.apps/expense-tracker-deployment created
```

```
PS D:\Indra\Programming\seleksi-lab\seleksi-sister\14.I-Am-Dripping> kubectl apply -f service.yml
service/expense-tracker-service created
```

4. Last one, jalankan command

- minikube service <nama_service>

command di atas digunakan untuk memulai service dalam kasus saya yaitu

- minikube service expense-tracker-service

```
PS C:\Users\DELL> minikube service expense-tracker-service
```

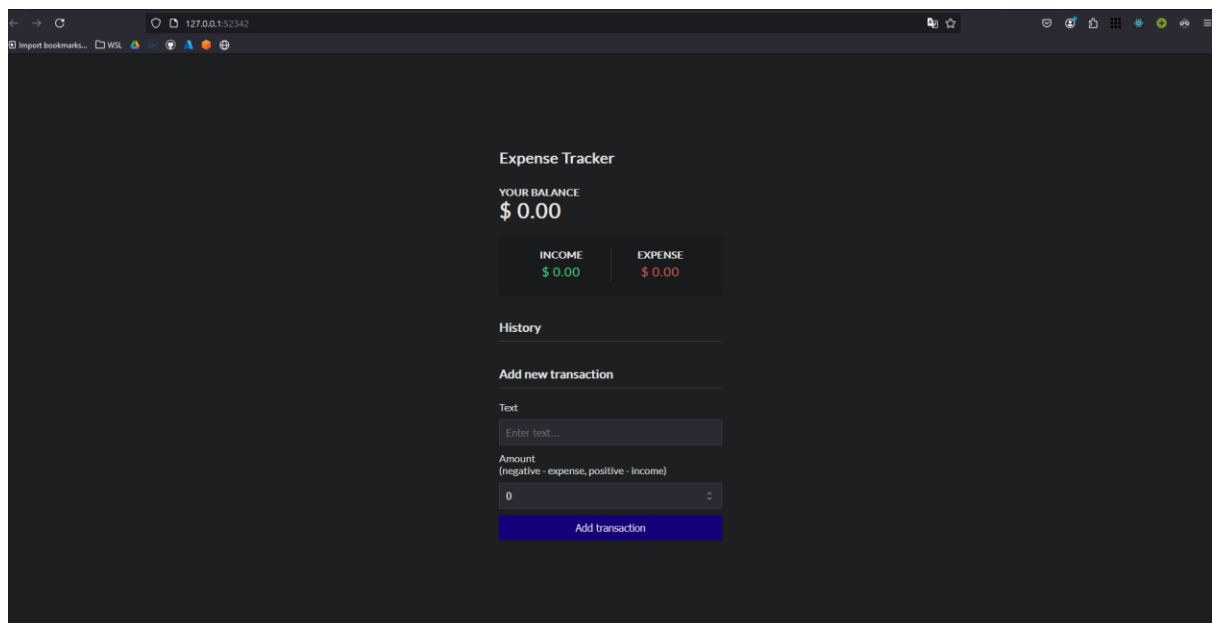
NAMESPACE	NAME	TARGET PORT	URL
default	expense-tracker-service	80	http://192.168.49.2:30000

Starting tunnel for service expense-tracker-service.

NAMESPACE	NAME	TARGET PORT	URL
default	expense-tracker-service		http://127.0.0.1:52342

Opening service default/expense-tracker-service in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.

tada, hasilnya bisa dilihat pada URL yang diberikan, dan ini hasilnya.



Note: saya pake webapp yang sama kayak soal Pelabuhan bedanya Pelabuhan 3 layer, yang ini gausah deh saya dah pusing kena azure kemaren.

TERIMA KASIH