Cara Launch FastAPI Service pada Containers Azure

Nama : Carissa Tabina Rianda

NIM : 18221120

1. Buatlah FastAPI App

FastAPI app dibuat dengan core bisnis yang telah dibuat pada tugas sebelumnya. Di sini saya menggunakan penjualan MealKit sebagai core dari bisnis saya. Core ini akan mengambil data user dan data barang yang akan dibeli, lalu mendata pemesanan sesuai ketersediaan barang. FastAPI App ini terdiri dari data yang dimasukkan dengan format JSON dan kode dengan Bahasa Python. Library yang digunakan adalah FastAPI dan Uvicorn

Cara Run Via Virtual Enviroment (venv) Python -- Windows

1. Pull repository ini ke dalam local folder
2. Buka terminal di VS Code atau via command prompt bisa
3. Ketik python -m venv [nama virtual enviroment (dibebaskan)] lalu enter
4. Masuk ke venv dengan cara ketik [nama virtual enviroment (dibebaskan)]\Scripts\activate
5. Install library terkait. Disini saya menggunakan fastapi dan uvicorn dengan cara pip install fastapi uvicorn
6. Jalankan aplikasi dengan uvicorn pemesanan:app --port 8000 --reload

```python
#Melakukan post ke JSON hasil_pemesanan
@app.post('/pemesanan')
async def add_hasil_pemesanan(item: InputUser):
    data = item.dict()
    i = 1
    jumlah = data.get("jumlah")
    barang_found = False
    stok_tersedia = False
    id_barang = data.get("id_barang")
    for barang in data_barang['data_barang']:
        if barang['id_barang'] == data['id_barang']:
            barang_found = True
            if barang['stok'] >= jumlah:
                stok_tersedia = True
                hasilPemesanan = "Anda berhasil memesan barang"
                nominal = jumlah * barang['harga']
                stok = barang['stok'] - jumlah
            else :
                hasilPemesanan = "Anda tidak berhasil memesan barang"
            break

    if not barang_found :
        return "Barang tidak tersedia"
    if not stok_tersedia :
        return "Anda tidak berhasil memesan barang"


# Update stok barang di dalam data_barang.json
    for barang in data_barang['data_barang']:
        if barang['id_barang'] == data['id_barang']:
            barang['stok'] -= jumlah
```

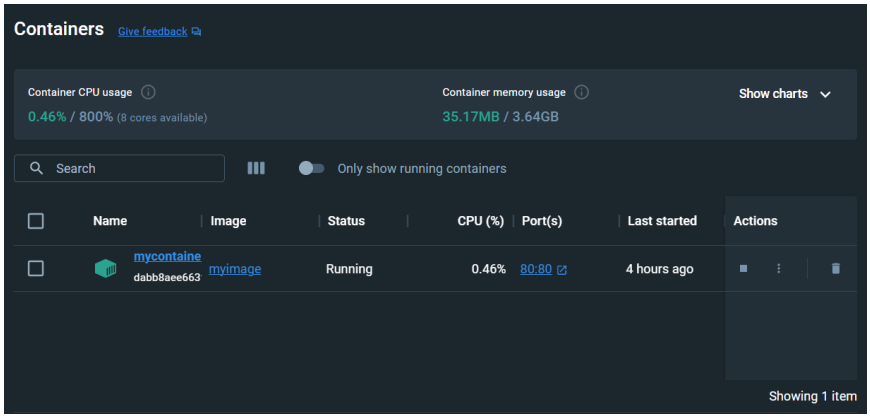
2. Dockerize aplikasi

Setelah aplikasi dibuat, buat DockerFile yang berisi kode berikut

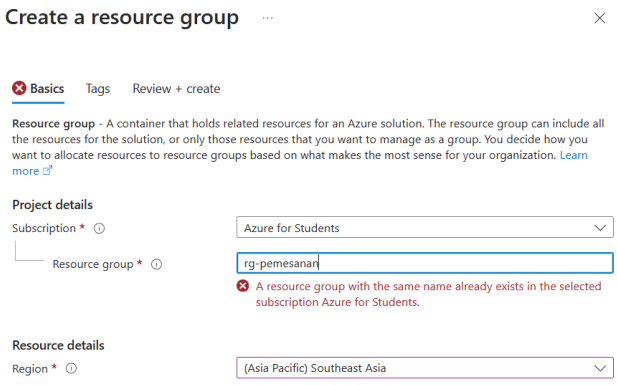
```
FROM python:3

ADD pemesanan.py .

COPY . /PEMESANANMEALKIT
WORKDIR /PEMESANANMEALKIT
RUN pip install fastapi uvicorn
CMD ["uvicorn", "pemesanan:app", "--host=0.0.0.0", "--port=80"]
```

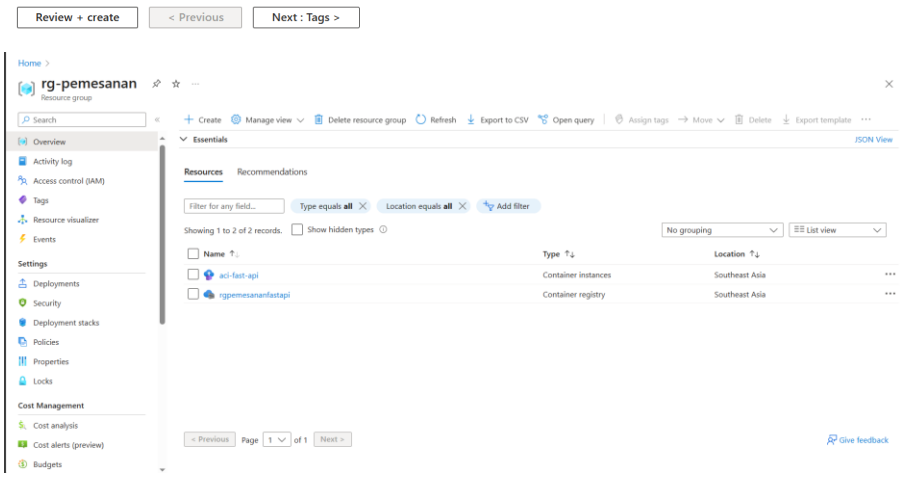
Lalu, build docker menggunakan docker build -t [tag docker (bebas)] .

Untuk run docker gunakan docker run -d --name mycontainer -p 80:80 [tag docker (bebas)]

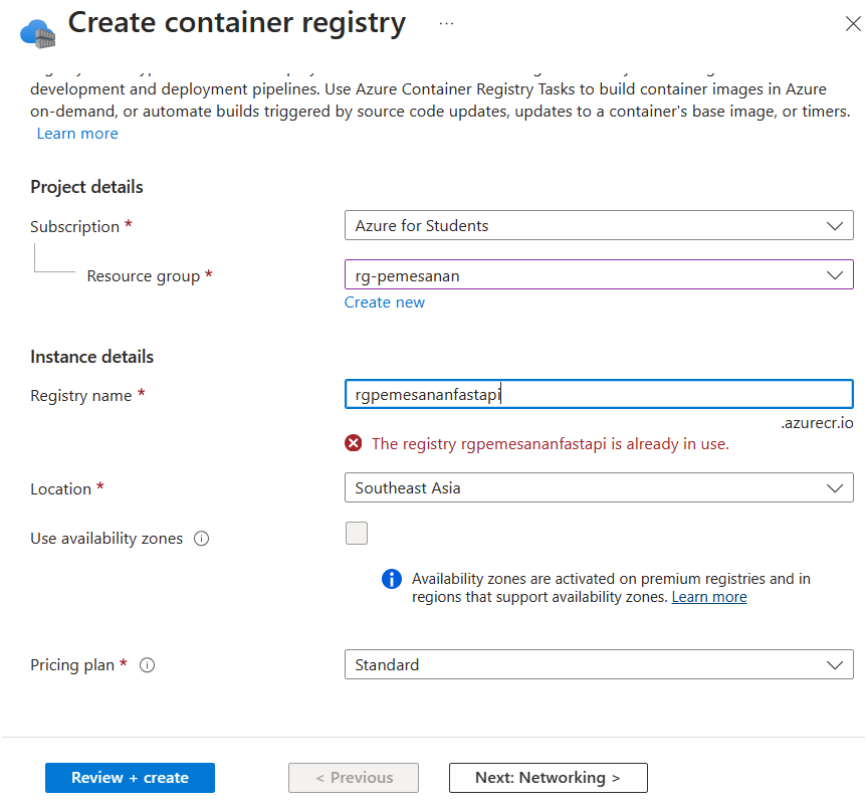


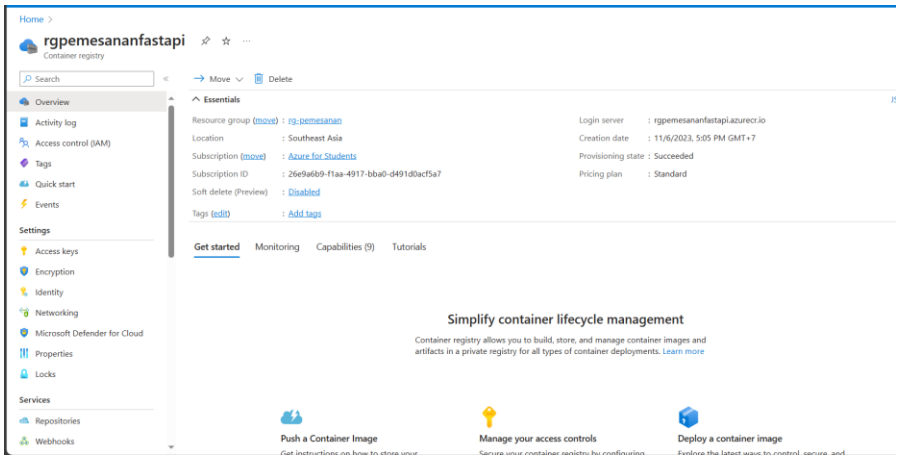
3. Buat Resource Group pada Azure





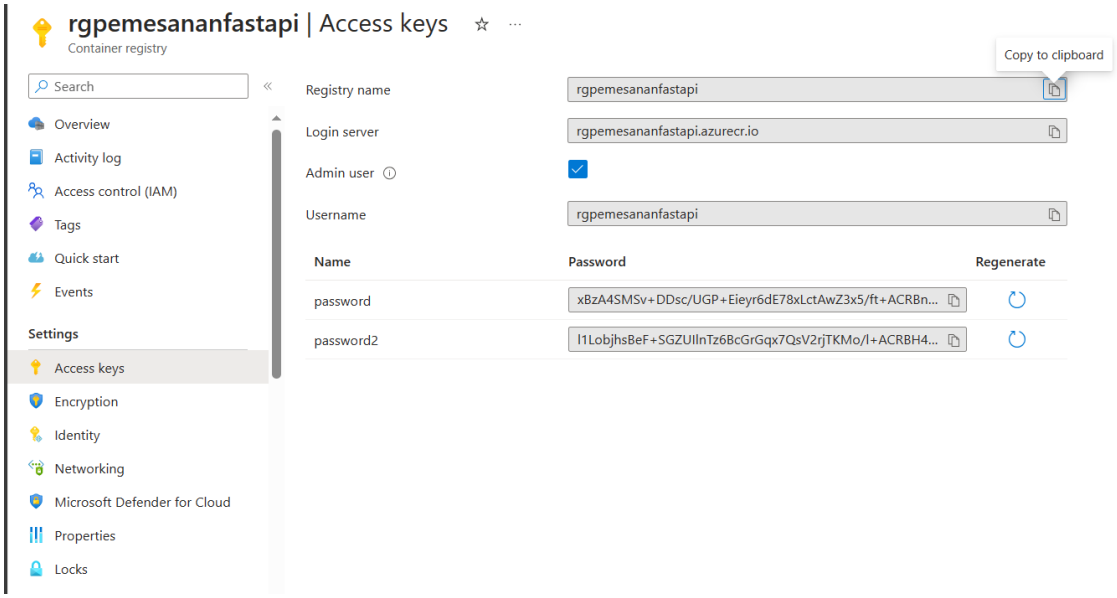
4. Buat Container Registry pada Azure





5. Ganti Access Keys pada Container Registry

Penggantian access key pada container registry digunakan untuk menghubungkan Docker dengan Container Registry



Ketik docker login rgpemesananfastapi.azurecr.io -u rgpemesananfastapi -p xBzA4SMSv+DDsc/UGP+Eieyr6dE78xLctAwZ3x5/ft+ACRBn+6Vr pada terminal

```
See 'docker run --help'.
PS D:\PemesananMealKit\TST> docker run -d --name mycontainer -p 80:80 myimage
c78cfd3694defd4744433184fe10fce75c724596a6e8574e7a26a406da410ce1
PS D:\PemesananMealKit\TST> docker login rgpemesananfastapi.azurecr.io -u rgpem
esananfastapi -p xBzA4SMSv+DDsc/UGP+Eieyr6dE78xLctAwZ3x5/ft+ACRBn+6Vr
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
PS D:\PemesananMealKit\TST>
```

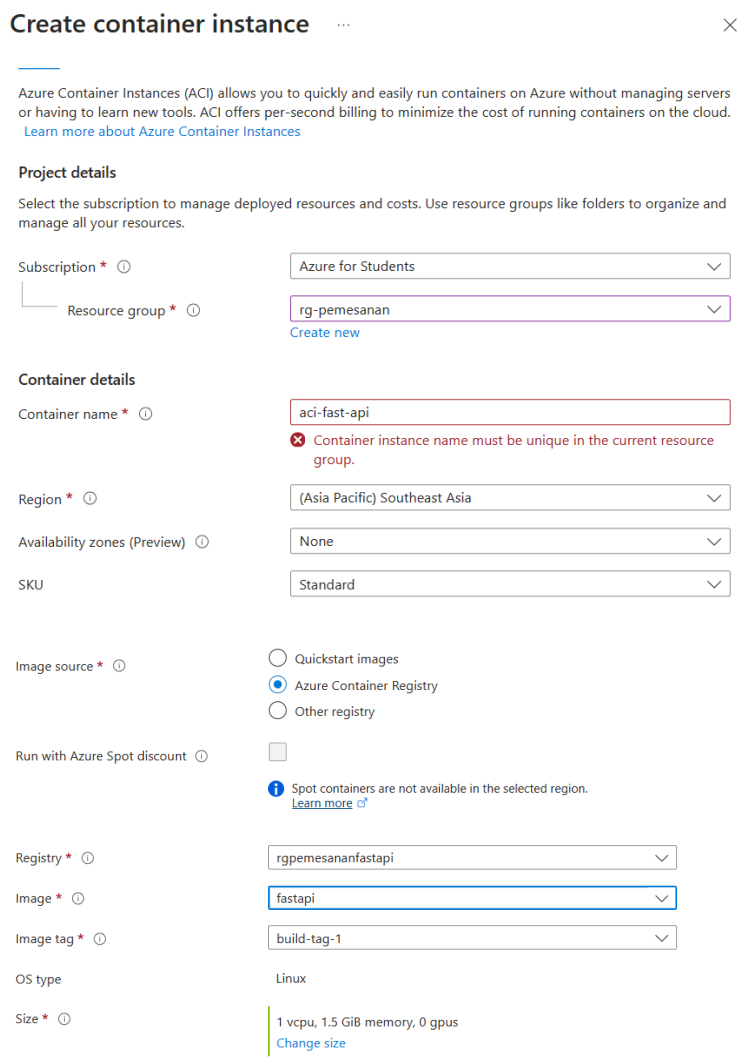
Lalu ketik docker build -t rgpemesananfastapi.azurecr.io/rgfastapi:build-tag-1 .

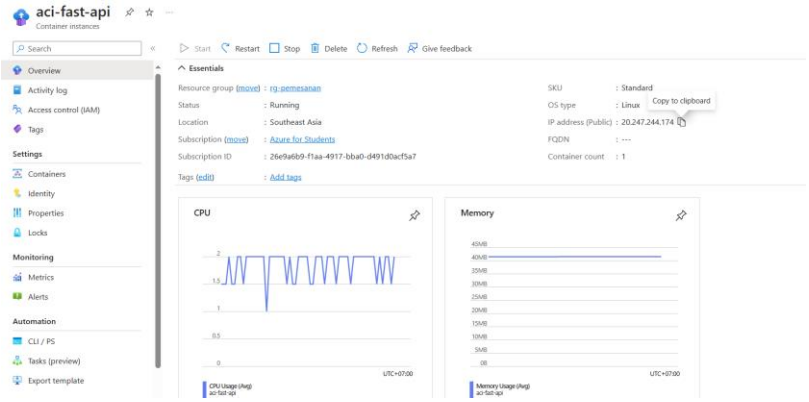
```
PS D:\PemesananMealKit\TST> docker build -t rgpemesananfastapi.azurecr.io/rgfas
tapi:build-tag-1 .
[+] Building 4.7s (11/11) FINISHED                              docker:default
 => [internal] load .dockerignore                                        0.0s
 => => transferring context: 2B                                          0.0s
 => [internal] load build definition from Dockerfile                     0.1s
 => => transferring dockerfile: 228B                                     0.0s
 => [internal] load metadata for docker.io/library/python:3              4.5s
 => [auth] library/python:pull token for registry-1.docker.io            0.0s
 => [1/5] FROM docker.io/library/python:3@sha256:7b8d65a924f596eb6530621  0.0s
 => [internal] load build context                                        0.0s
 => => transferring context: 267B                                        0.0s
 => CACHED [2/5] ADD pemesanan.py .                                      0.0s
 => CACHED [3/5] COPY . /PEMESANANMEALKIT                                0.0s
 => CACHED [4/5] WORKDIR /PEMESANANMEALKIT                               0.0s
 => CACHED [5/5] RUN pip install fastapi uvicorn                         0.0s
 => exporting to image                                                   0.0s
 => => exporting layers                                                  0.0s
 => => writing image sha256:0025b4a8aecfcea454fd17fce291f731fc7c8c3c4a0b  0.0s
 => => naming to rgpemesananfastapi.azurecr.io/rgfastapi:build-tag-1      0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout qu
ickview
PS D:\PemesananMealKit\TST>
```

Docker push rgpemesananfastapi.azurecr.io/rgfastapi:build-tag-1

```
74c0c15e1c9dc size: 2841
PS D:\PemesananMealKit\TST> docker push rgpemesananfastapi.azurecr.io/rgfastapi
:build-tag-1
The push refers to repository [rgpemesananfastapi.azurecr.io/rgfastapi]
41213a474334: Layer already exists
5f70bf18a086: Layer already exists
579eb899aa27: Layer already exists
9d6209a4511c: Layer already exists
701d0b971f5f: Layer already exists
619584b251c8: Layer already exists
ac630c4fd960: Layer already exists
86e50e0709ee: Layer already exists
12b956927ba2: Layer already exists
266def75d28e: Layer already exists
29e49b59edda: Layer already exists
1777ac7d307b: Layer already exists
build-tag-1: digest: sha256:bc0afcfda56d048ce75f009a6abe3e074bdec9901a7b93414b4
74c0c15e1c9dc size: 2841
```

6. Buat Container Instance





7. Menjalankan Container

Akses layanan FastAPI menggunakan IP address **http://20.247.244.174/docs**