

# Pelabuhan

(TuT TuT TuT)

## Akses Cloud

Untuk services yang saya gunakan adalah *Microsoft azure*. Pake ini karena dapet packet student yang free. Berikut Langkah-langkah yang perlu dilakukan untuk claim(?)

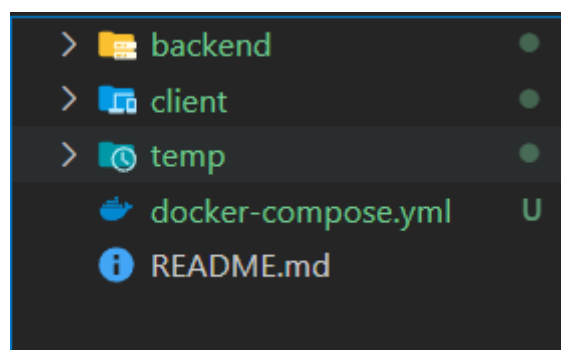
1. Kunjungi URI ini <https://azure.microsoft.com/id-id/free/students>
2. Klik mulai gratis dan sesuaikan dengan akun klen.
3. Harusnya nanti nunggu konfirmasi dulu dan dapet deh azure student.

## Dockerize

Pertama-tama untuk website yang akan di-docker saya menggunakan website abal-abal yang pernah saya buat pas liburan, yaitu website ini <https://github.com/Indraswara/expense-tracker>. Struktur directory-nya kurang lebih begini

Indraswara	Update README.md	412f803 · last month	11 Commits
client	finish	last month	
config	initialize backend	last month	
controllers	initialize backend	last month	
models	initialize backend	last month	
routes	initialize backend	last month	
.gitignore	server	last month	
README.md	Update README.md	last month	
package-lock.json	server	last month	
package.json	server	last month	
server.js	initialize backend	last month	

Nanti, ini akan dilakukan sedikit perubahan sehingga menjadi seperti ini.



Abaikan folder temp, karena itu buat docker-compose.yml untuk deploy nanti. Ketika sudah seperti ini kita bikin file “dockerfile” dan “.dockerignore” pada directory client (frontend) dan directory backend.

#### Dockerfile client

```
# client/Dockerfile

#membuat image dari node versi 20.15.0
FROM node:20.15.0

#membuat direktori /app
WORKDIR /app

#copy package.json dan package-lock.json ke direktori /app
COPY package*.json ./

#install dependensi
RUN npm install

#copy semua file ke direktori /app
COPY . .

#expose port 3000 untuk mengakses aplikasi
EXPOSE 3000

#menjalankan webapp
CMD ["npm", "start"]
```

#### Dockerfile backend

```
# backend/Dockerfile

#ini untuk membuat image dari node versi 20.15.0
FROM node:20.15.0

#membuat direktori /app
WORKDIR /app

#copy package.json dan package-lock.json ke direktori /app
COPY package*.json ./
#install dependensi
RUN npm install

#copy semua file ke direktori /app
COPY . .

#expose port 5000 untuk mengakses aplikasi
EXPOSE 5000

#menjalankan server.js
```

```
CMD ["node", "server.js"]
```

.Dockerignore untuk client dan backend isinya sama, yaitu hanya untuk mengabaikan directory node\_modules.

.Dockerignore

```
node_modules/
```

Lalu, kita buat docker-compose.yml pada root directory untuk menjalankan 2 container frontend dan backend secara bersamaan.

Docker-compose.yml

```
version: '3.8'

services:
  client:
    build:
      context: ./client
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    environment:
      - REACT_APP_BACKEND_URL=http://backend:5000
    depends_on:
      - backend

  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile
    ports:
      - "5000:5000"
    environment:
      - MONGO_URI=mongodb+srv://indra:indra@wallet.uilaiah.mongodb.net/expense_tracker?retryWrites=true&w=majority&appName=Wallet

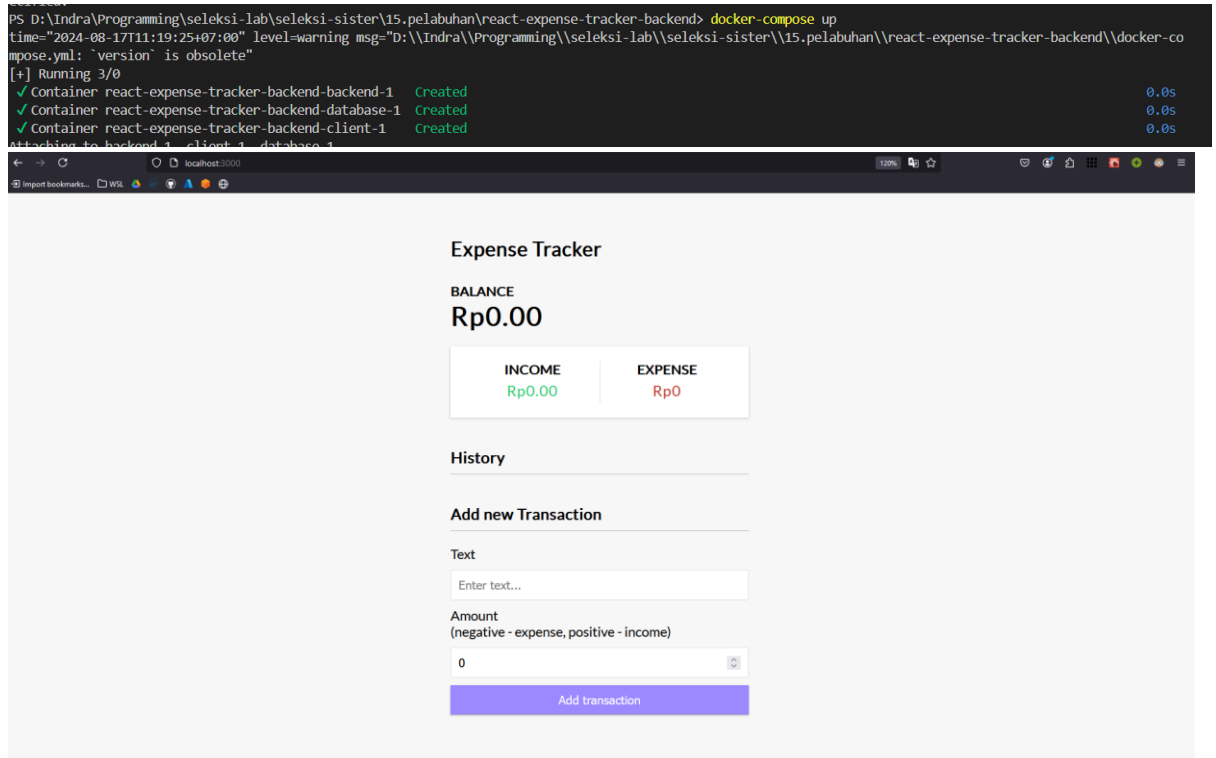
  database:
    image: mongo:latest
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db

volumes:
  mongo-data:
```

Kayanya docker ini 3 tier deh mungkin bonus? Gatau aku sendiri juga gayakin.

Setelah selesai kita bisa menjalankannya dengan command berikut pada terminal. **Kalo di Windows pastikan kalian membuka aplikasi docker desktop terlebih dahulu**

- Docker-compose build (untuk build container-nya)
- Docker-compose up (untuk menjalankan container-nya sehingga bisa webapp bisa diakses melalui localhost)



- Docker-compose down (untuk menonaktifkan container yang sedang menyala)

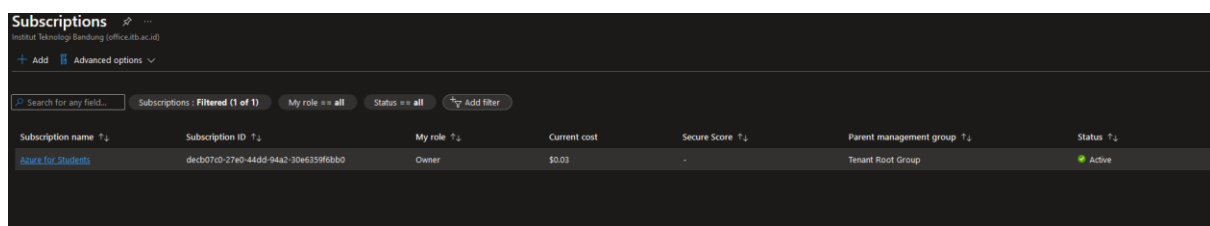
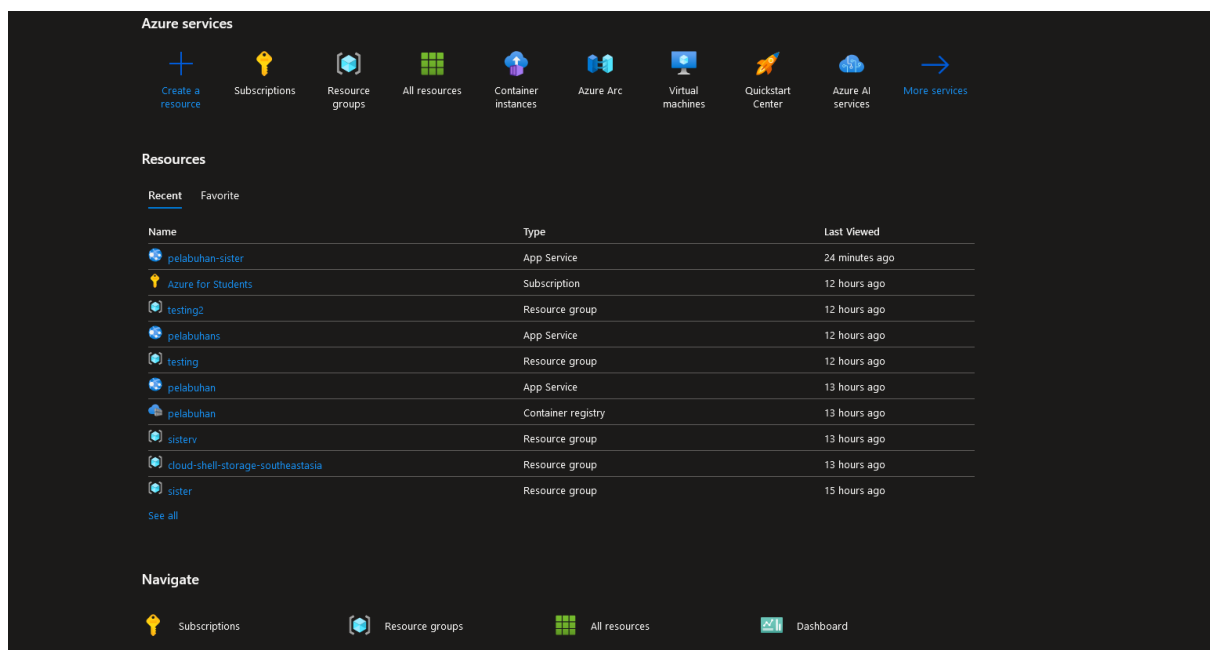
# Deployment

## Access Azure

Deployment disini saya gunakan azure karena dapat azure for students. Buat dapetannya tinggal kunjungi aja website ini

<https://azure.microsoft.com/id-id/free/students>

dan ikuti prosedurnya, nanti kalau udah selesai di beranda Microsoft azure bagian subscriptions kita bisa melihat subscriptions yang sudah kita dapat

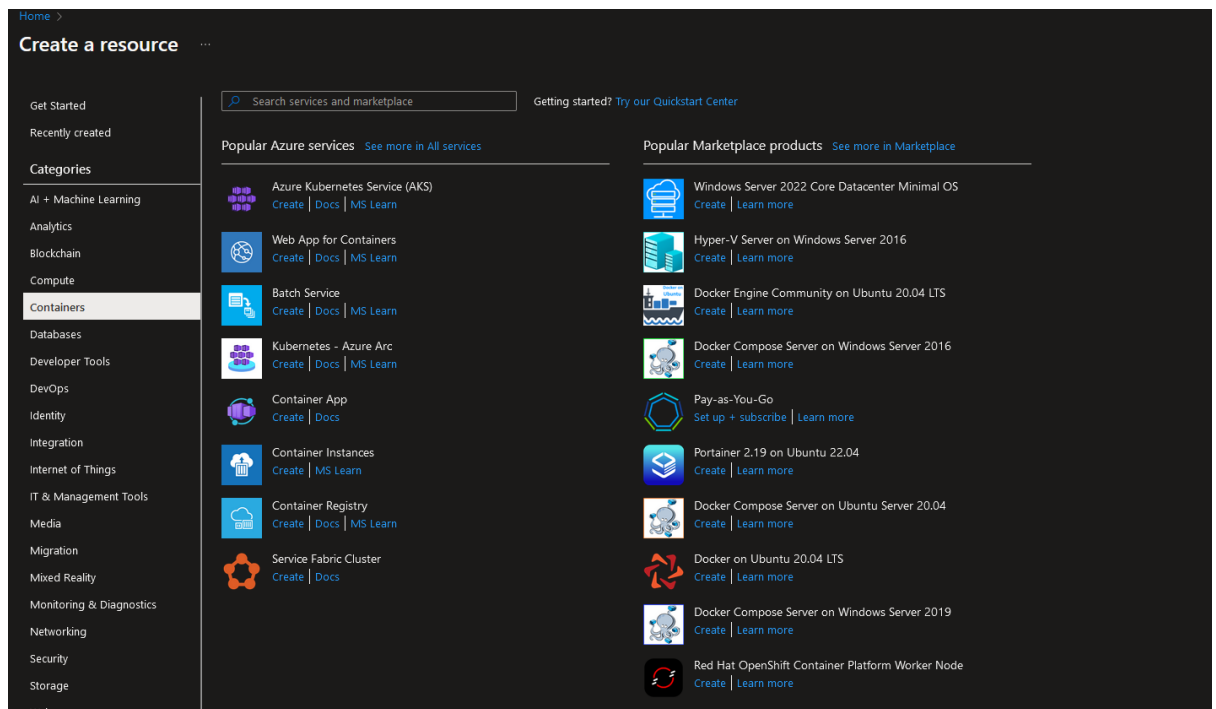


## Step By Step

### Bikin ACR (Azure Container Registry)

Bagian ini kita akan membuat ACR terlebih dahulu yang nantinya isinya adalah image dari webapp yang sudah dibuat, nantinya akan ada tiga, yaitu frontend, backend, dan database. Cara bikinnya

1. Klik Create Resource
2. Pilih Container Registry



### 3. Sisanya ini bagian Basics aja terus langsung review and create

Home > Create a resource >

## Create container registry

**Basics** Networking Encryption Tags Review + create

Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. Use Azure container registries with your existing container development and deployment pipelines. Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers. [Learn more](#)

**Project details**

Subscription \* Azure for Students

Resource group \* Create new

**Instance details**

Registry name \* Enter the name .azurecr.io

Location \* Southeast Asia

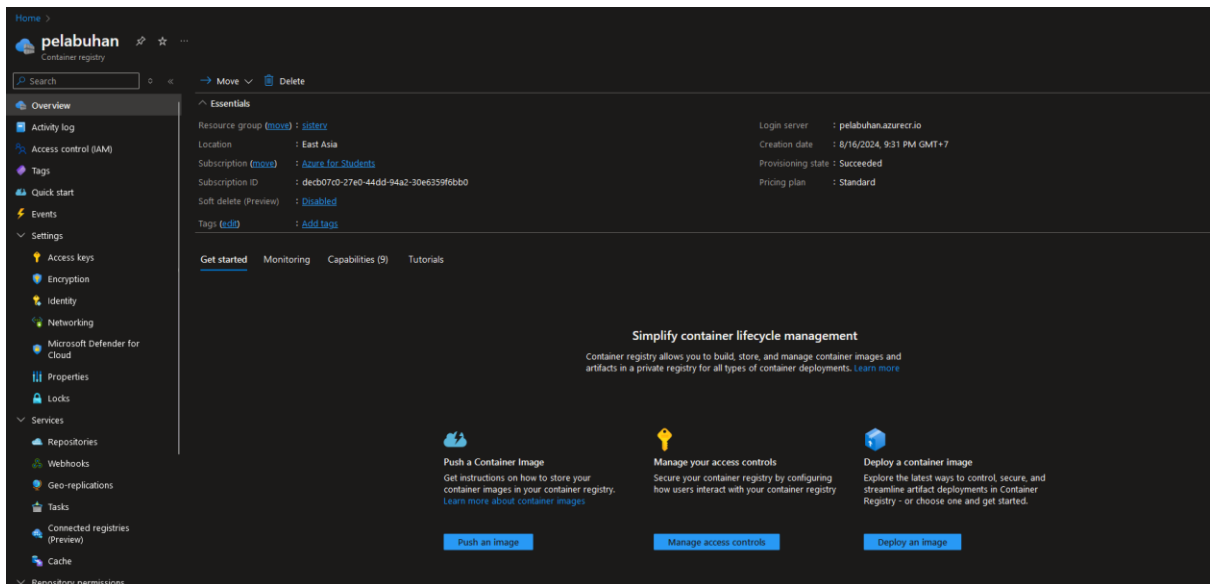
Use availability zones ☐

**Pricing plan** \* Standard

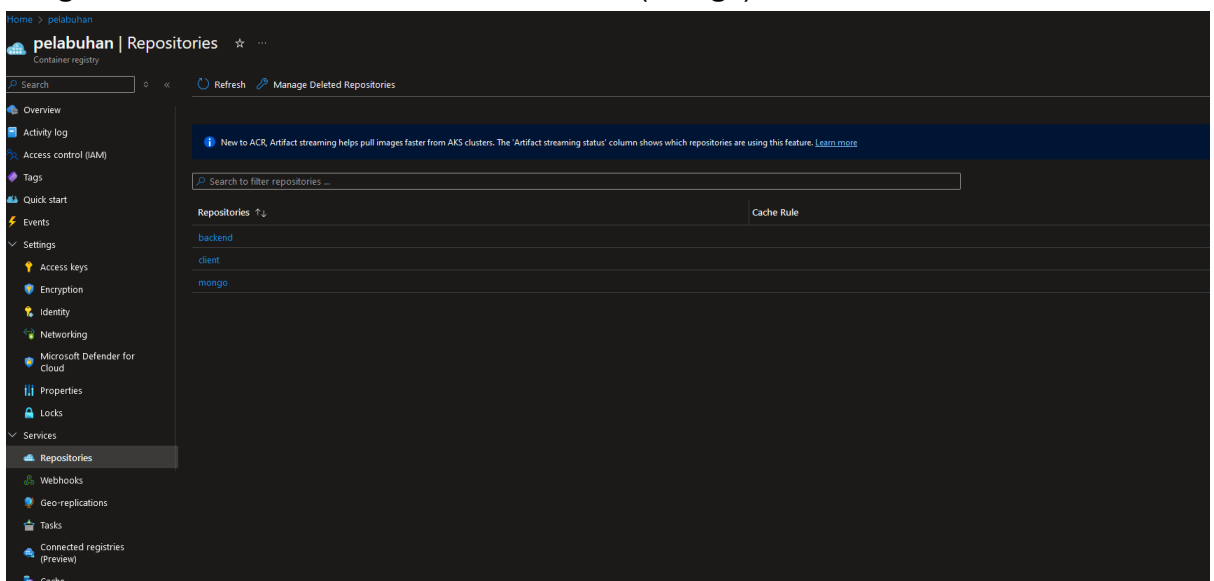
[Availability zones are activated on premium registries and in regions that support availability zones. \[Learn more\]\(#\)](#)

[Review + create](#) [< Previous](#) [Next: Networking >](#)

### 4. Kalo udah di-deploy harusnya nanti kita bisa memiliki container registry seperti berikut



5. Nanti kita bisa push image kita ke registry ini dengan mengikuti prosedur “push a container image”. Hasil dari push tadi bisa kita lihat di bagian repositories. Pada gambar di bawah terlihat 3 repositories yang masing-masing adalah container image untuk frontend, backend, dan database(mongo).

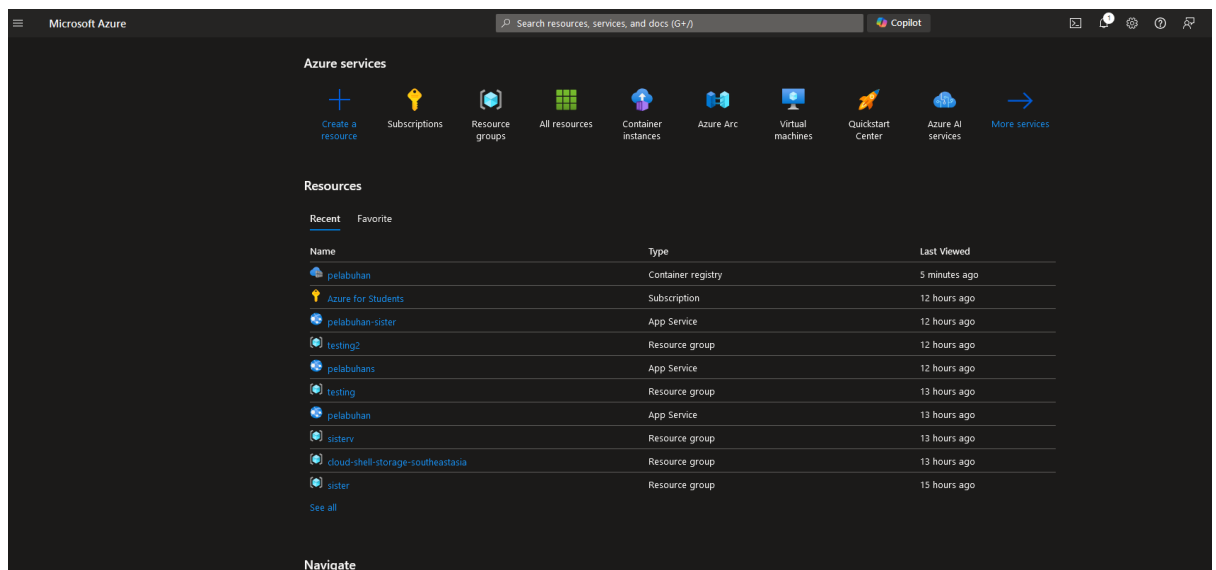


6. Bagian ini sudah selesai dan kita bisa lanjut untuk deploy webapp-nya

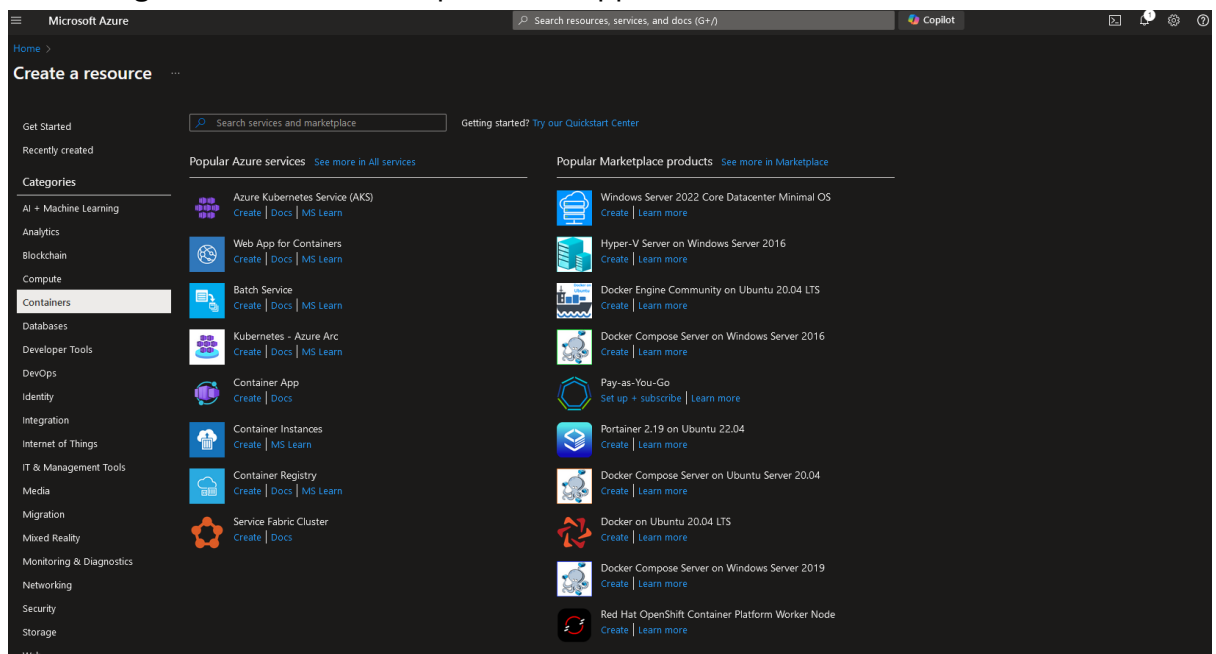
## Deploy Webapp

1. Pilih create resource pada home





## 2. Pilih categories containers dan pilih “web app for containers”



## 3. Bagian basics bisa diisi sesuai kebutuhan dan jangan lupa bagian publish pilih container

**Project Details**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ Azure for Students

Resource Group \* ⓘ demo

[Create new](#)

**Instance Details**

Name \* pelabuhan

-fjcqbqdevanbedv.eastus-01.azurewebsites.net

☒ Unique default hostname (preview) on. [More about this update](#)

Publish \* ☐ Code ☒ Container ☐ Static Web App

Operating System \* ☒ Linux ☐ Windows

Region \* East US

**Pricing plans**

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Linux Plan (East US) \* ⓘ ASP-demo-9c99 (B1)

[Create new](#)

Pricing plan **Basic B1** (100 total ACU, 1.75 GB memory, 1 vCPU)

- Selanjutnya ke bagian docker, nanti disini pilih docker compose (karena container lebih dari satu). Image Resource pilih ACR, Registrynya pilih registry yang kita pakai untuk menyimpan image di awal tadi. Terakhir, untuk configuration masukkan docker-compose.yml

Basics Deployment **Docker** Container Networking Monitor + secure Tags Review + create

Pull container images from Azure Container Registry, Docker Hub or a private Docker repository. App Service will deploy the containerized app with your preferred dependencies to production in seconds.

Options Docker Compose (Preview)

Image Source Azure Container Registry

**Azure container registry options**

Registry \* pelabuhan

Configuration File "docker.compose.yml"

Configuration ⓘ

```
version: '3.8'

services:
  client:
    image: pelabuhan.azurecr.io/client:latest
    ports:
      - "3000:3000"
```

Bagian docker.compose.yml berbeda dari docker.compose.yml pada dockerize disini berbeda

Berikut codenya,

```
version: '3.8'

services:
  client:
    image: pelabuhan.azurecr.io/client:latest
    ports:
      - "3000:3000"
    environment:
      - REACT_APP_BACKEND_URL=http://backend:5000
      - DANGEROUSLY_DISABLE_HOST_CHECK=true
    depends_on:
      - backend

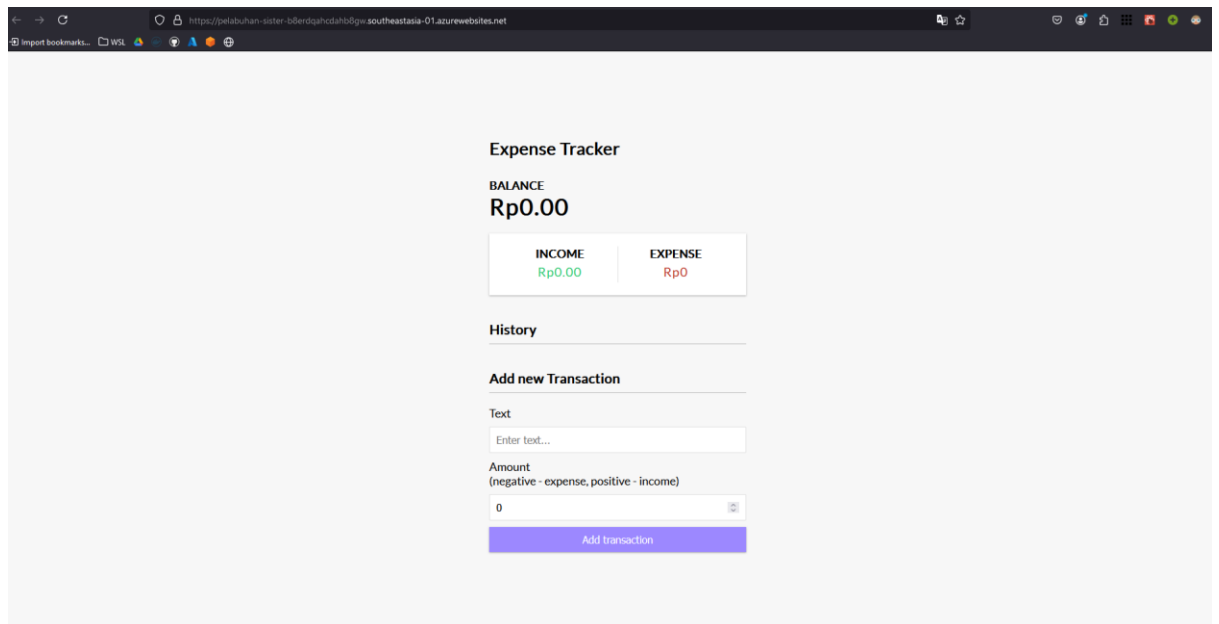
  backend:
    image: pelabuhan.azurecr.io/backend:latest
    ports:
      - "5000:5000"
    environment:
      - MONGO_URI=mongodb+srv://indra:indra@wallet.uilaiah.mongodb.net/expense_tracker?retryWrites=true&w=majority&appName=Wallet

  database:
    image: pelabuhan.azurecr.io/mongo:latest
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db

volumes:
  mongo-data:
```

Bagian image berbeda, yang kita masukkan adalah link untuk ke repositories yang ada di dalam registry.

5. Bagian Container isi sama seperti Docker
6. Terakhir kita bisa langsung review + create dan tunggu hasilnya.
7. Nah, untuk punya saya ini hasil deploy-nya  
<https://pelabuhan-sister-b8erdqahcdahb8gw.southeastasia-01.azurewebsites.net/>



8. Dan selesai (gweh habis 5-6 jam buat deploy ini anjim)