

PROJECT NETWORKING, SECURITY, and ORCHESTRATION DOCKER

Deskripsi :

Sebuah startup ingin membangun aplikasi berbasis web dengan arsitektur berikut:

1. Frontend: Menggunakan Nginx.
2. Backend: API sederhana berbasis Python.
3. Database: PostgreSQL untuk menyimpan data aplikasi.
4. Networking: Backend dan database harus saling terhubung dengan jaringan.
5. Volume: Database PostgreSQL memerlukan penyimpanan data yang persisten.

Tugas :

1. Buat Docker Compose file untuk menjalankan semua service.
 2. Pastikan Front End dapat diakses melalui browser.
 3. Gunakan Docker Volume untuk menyimpan data PostgreSQL.
 4. Pastikan semua container saling terhubung melalui jaringan.
-

1. Buat Docker Compose file untuk menjalankan semua service :

```
version: '3.8'

services:
  frontend:
    build: ./frontend
    ports:
      - "8080:80"
    networks:
      - app-network

  backend:
    build: ./backend
    ports:
      - "5001:5000" # Port 5001 di host
    environment:
      - FLASK_ENV=development
      - DATABASE_URL=postgresql://appuser:password@db:5432/appdb
    networks:
      - app-network

  db:
    image: postgres:13
    environment:
      - POSTGRES_USER=appuser
      - POSTGRES_PASSWORD=password
      - POSTGRES_DB=appdb
    volumes:
      - db-data:/var/lib/postgresql/data
    networks:
      - app-network

volumes:
  db-data:

networks:
  app-network:
    driver: bridge
```

2. Pastikan Front end dapat diakses melalui browser :

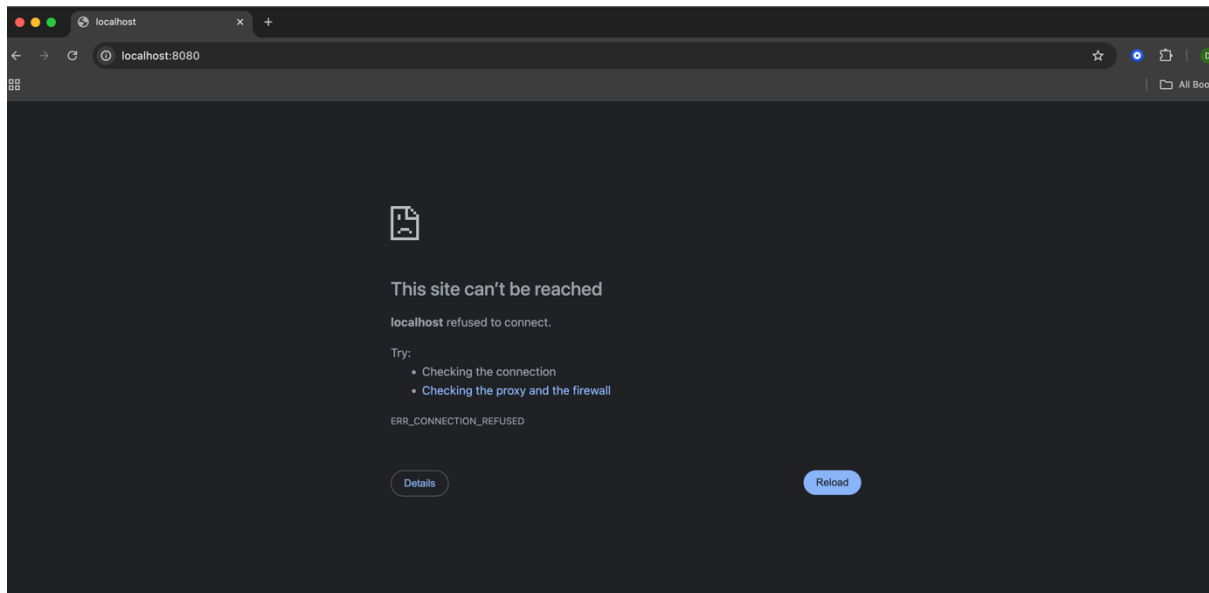
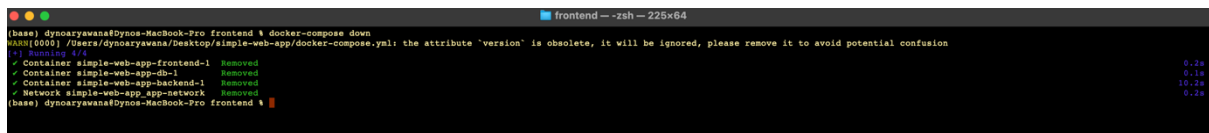
(menjalankan Docker Compose dengan *docker-compose up -d --build*)

```
frontend -- zsh -- 225x64
(base) dynoaryawana@Dynos-MacBook-Pro frontend % docker-compose up -d --build
WARN[0000] /Users/dynoaryawana/Desktop/simple-web-app/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 1.4s (21/21) FINISHED
=> [frontend internal] load build definition from Dockerfile
=> transferring dockerfile: 132B
=> [backend internal] load build definition from Dockerfile
=> transferring dockerfile: 167B
=> [backend internal] load metadata for docker.io/library/python:3.9-slim
=> [frontend internal] load metadata for docker.io/library/nginx:alpine
=> [frontend internal] load .dockerignore
=> transferring context: 2B
=> [frontend 1/3] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
=> resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
=> [frontend internal] load build context
=> transferring context: 61B
=> CACHED [frontend 2/3] COPY index.html /usr/share/nginx/html/
=> CACHED [frontend 3/3] COPY style.css /usr/share/nginx/html/
=> [frontend] exporting to image
=> exporting layers
=> exporting manifest sha256:42449e1eb1149418a1708ba639a543daed186f5ed9fda849d6e46a87b0a1b8
=> exporting config sha256:282bd2299b1f404d1c417a917ed18f80b256e980f783539fc2fa3e1a741d02a6
=> exporting attestation manifest sha256:d8f4a239df76ecf39e748ca374b039befd122da055ea7282114b5cbcea7671c
=> exporting manifest list sha256:49e354eaa5ab388ef80c4b7914fa95a1d73986a5f0f08f67dc72365e965bebe
=> naming to docker.io/library/simple-web-app-frontend:latest
=> unpacking to docker.io/library/simple-web-app-frontend:latest
[backend auth] library/python:pull token for registry-1.docker.io
=> [frontend] resolving provenance for metadata file
=> [backend internal] load .dockerignore
=> transferring context: 2B
=> [backend 1/5] FROM docker.io/library/python:3.9-slim@sha256:aff2066ec8914f7383e115bbcded4d24da428eac377b0d4bb73806de992d240f
=> resolve docker.io/library/python:3.9-slim@sha256:aff2066ec8914f7383e115bbcded4d24da428eac377b0d4bb73806de992d240f
=> [backend internal] load build context
=> transferring context: 93B
=> CACHED [backend 2/5] WORKDIR /app
=> CACHED [backend 3/5] COPY requirements.txt .
=> CACHED [backend 4/5] RUN pip install -r requirements.txt
=> CACHED [backend 5/5] COPY . .
=> [backend] exporting to image
=> exporting layers
=> exporting manifest sha256:a89d5c3ac020849d6485e4e91183d992f4a343d606282ff954174d2a70b86725
=> exporting config sha256:4909bfe62d740a25f57c17bdd362d6d805a59ab9e221c0d1b4f69d3196e23f2
=> exporting attestation manifest sha256:a90a7ed680542af1914df14c4bd6d56d1796ad146d25521d31fed44a8c39cf47
=> exporting manifest list sha256:c67d983367cc6dabef73d51761c8bc81baa2c3f90c8e9450b64e8f6b06f926a2
=> naming to docker.io/library/simple-web-app-backend:latest
=> unpacking to docker.io/library/simple-web-app-backend:latest
=> [backend] resolving provenance for metadata file
[+] Running 6/6
✔ backend                               Build
✔ frontend                               Build
✔ Network simple-web-app_app-network    Created
✔ Container simple-web-app-backend-1    Started
✔ Container simple-web-app-db-1         Started
✔ Container simple-web-app-frontend-1   Started
(base) dynoaryawana@Dynos-MacBook-Pro frontend %
```

(mengakses website dengan <http://localhost:8080/> di web browser)



(mengecek server apakah berjalan dengan mematikan dan menghidupkan kembali server dengan ***docker-compose down*** dan ***docker-compose up -d --build***).



3. Mengecek apakah Docker Volume nya sudah terbuat utk PostgreSQL dengan *docker volume ls* dan *docker volume inspect nama_docker_volume*

```
(base) dnyoaryawana@Dynos-MacBook-Pro frontend % docker volume inspect simple-web-app_db-data
[
  {
    "CreatedAt": "2025-05-22T18:44:59Z",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.config-hash": "9e2806d2cbcd586ed2b81400df6a20cb4aac22e5eca16806404ab748625183c",
      "com.docker.compose.project": "simple-web-app",
      "com.docker.compose.version": "2.34.0",
      "com.docker.compose.volume": "db-data"
    },
    "Mountpoint": "/var/lib/docker/volumes/simple-web-app_db-data/_data",
    "Name": "simple-web-app_db-data",
    "Options": null,
    "Scope": "local"
  }
]
(base) dnyoaryawana@Dynos-MacBook-Pro frontend %
```

4. Pastikan semua container saling terhubung melalui jaringan dengan : *docker network inspect nama_docker_network*

```
(base) dnyoaryawana@Dynos-MacBook-Pro frontend % docker network inspect simple-web-app_app-network
[
  {
    "Name": "simple-web-app_app-network",
    "Id": "2c1b997f26172c7947273d87d545c946c424dca90385c039ff800047981eb14",
    "Created": "2025-05-22T20:43:28.987110178",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.19.0.0/16",
          "Gateway": "172.19.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "06368326a29cbdd7cb992e53188b1b51e7c10a31e5d8cacebc268dfec4b04aa": {
        "Name": "simple-web-app-backend-1",
        "EndpointID": "5c5a9765f15767aac2209b422d3bcf261005e48f532c2cad1cffe8bf127b902a",
        "MacAddress": "16:3a:41:d4:5f:fb",
        "IPv4Address": "172.19.0.4/16",
        "IPv6Address": ""
      },
      "62ceab4ee376d10fc070eb266a6350fbc4dcf94d2e9661dd2a3ba61fd7b44da2": {
        "Name": "simple-web-app-frontend-1",
        "EndpointID": "9e6d9f04d2237c9bae4f733883d91a40863dbfcbad10ald4e5d59cdd4e9e892",
        "MacAddress": "0e:5b:09:09:81:72",
        "IPv4Address": "172.19.0.2/16",
        "IPv6Address": ""
      },
      "bc023510eac7c0856bd4ccd30d3efa7c393685615c9ad6f7917ceb4315c31662": {
        "Name": "simple-web-app-db-1",
        "EndpointID": "01972a850a50dc08bc4f33401796ada804af37dee8c186d4347af3b5c9978666",
        "MacAddress": "42:f8:ff:13:e8:26",
        "IPv4Address": "172.19.0.3/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {
      "com.docker.compose.config-hash": "041731e6f6dd2e0052b74c6643db1bd943cfff2f4ec38bc4d113fafd9890f3e8b",
      "com.docker.compose.network": "app-network",
      "com.docker.compose.project": "simple-web-app",
      "com.docker.compose.version": "2.34.0"
    }
  }
]
```

Thank You