

## Flask-Redis uygulaması

### 1. Docker Ağı Oluşturma

`mynet` adında bir Docker ağı oluşturun:

```
docker network create mynet
```

### 2. Redis Veritabanı Konteyneri Çalıştırma

`redis` Docker imajını kullanarak Redis veritabanını dağıtmak için bir konteyner çalıştırın:

```
docker run --name mydb --network mynet -d redis
```

Bu konteyner, ziyaret sayısını kaydetmek için gerekli olan önbellek verilerini saklayacak Redis veritabanına sahip olacaktır.

### 3. Flask Uygulama Kodunu Yazma

`app.py` adında bir dosya oluşturun ve aşağıdaki Python kodunu içine yazın. Bu kod, Flask uygulaması tarafından barındırılan sitedeki ziyaret sayısını güncellemek için kullanılacaktır:

```
from flask import Flask
from redis import Redis

app = Flask(__name__)
redis = Redis(host='mydb', port=6379)

@app.route('/')
def hello():
    count = redis.incr('hits')
    return f'Hello! This site has been visited {count} times.'

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

### 4. Gereksinimler Dosyası Oluşturma

`requirements.txt` adında bir dosya oluşturun ve içine aşağıdaki satırları ekleyin:txt

```
flask
redis
```

## 5. Dockerfile Oluşturma

Flask uygulaması için aşağıdaki talimatlarla bir Dockerfile oluşturun:

```
FROM python:3.8-slim

WORKDIR /app

COPY requirements.txt requirements.txt

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]
```

## 6. Docker İmajını Oluşturma

Yukarıdaki Dockerfile'ı kullanarak bir Docker imajı oluşturun:

```
docker build -t flask .
```

## 7. Flask Uygulamasını Default Bridge Network'te Çalıştırma ve Redis Bağlantı Sorununu Gösterme

Flask uygulamasını varsayılan bridge network'te çalıştırın ve Redis veritabanına bağlanamayacağını gözlemleyin:

```
docker run --name App1 -p 5000:5000 flask
```

Bu komut ile Flask uygulaması varsayılan Docker ağında çalışacak ve **mydb** adındaki Redis veritabanını bulamayacak. Flask uygulaması çalışırken Redis'e bağlantı kuramadığına dair hata mesajları alacaksınız.

```
(base) redis-flask ➦ docker run --name App1 -p 5000:5000 flask
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.215.2:5000
Press CTRL+C to quit
[2024-08-20 12:52:20,816] ERROR in app: Exception on / [GET]
Traceback (most recent call last):
  File "/usr/local/lib/python3.8/site-packages/redis/connection.py", line 277, in connect
    sock = self.retry.call_with_retry(
  File "/usr/local/lib/python3.8/site-packages/redis/retry.py", line 62, in call_with_retry
    return do()
  File "/usr/local/lib/python3.8/site-packages/redis/connection.py", line 278, in <lambda>
    lambda: self._connect(), lambda error: self.disconnect(error)
  File "/usr/local/lib/python3.8/site-packages/redis/connection.py", line 607, in _connect
    for res in socket.getaddrinfo(
  File "/usr/local/lib/python3.8/socket.py", line 918, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
socket.gaierror: [Errno -2] Name or service not known
```

## 8. Flask Uygulamasını Durdurup Network'ünü Değiştirme

Şimdi Flask uygulamasını durdurup, network'ünü **mynet** olarak değiştirelim ve Redis'e bağlanmasını sağlayalım:

**Flask konteynerini durdurun:**

```
docker stop App1
```

Flask konteynerini mevcut network'ten ayırın:

```
docker network disconnect bridge App1
```

Flask konteynerini **mynet** ağına bağlayın:

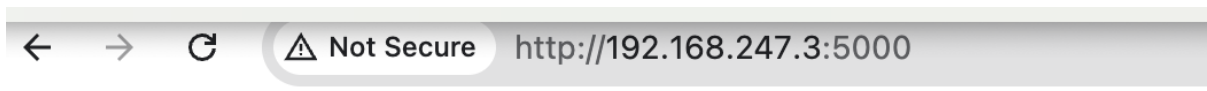
```
docker network connect mynet App1
```

Flask konteynerini yeniden başlatın:

```
docker start App1
```

```
(base) docker ➦ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
38404e129ad8   flask    "python app.py"         About a minute ago   Up About a minute   0.0.0.0:5000->5000/tcp
25c9b7f0a111   redis    "docker-entrypoint.s... About a minute ago   Up About a minute   6379/tcp
(mydb)
(base) docker ➦ docker stop App1
App1
(base) docker ➦ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
2562ba72c540   bridge   bridge    local
cb0885d312c2   host     host      local
4802245869a8   mynet    bridge    local
a009d670231b   none     null      local
(base) docker ➦ docker network disconnect bridge App1
(base) docker ➦ docker network connect mynet App1
(base) docker ➦ docker start App1
App1
```

Bu adımların ardından Flask uygulaması Redis veritabanına bağlanabilecek ve ziyaret sayısını güncelleyebilecektir.



Hello! This site has been visited 4 times.