

lab01 - Środowisko mikrouslug

1. Początkowo zadania zacząłem robić na systemie MAC OS, ale pojawił się problem z `--network host`. Mimo, że serwer był utworzony prawidłowo to połączyć się mogłem tylko z wewnątrz kontera. Znalazłem w internecie jakieś rozwiązania typu bridge, ale nie chciałem tracić czasu i przenieśliem się na Ubuntu.

Program Visual Studio Code

2. Zainstaluj w środowisku utworzonej maszyny wirtualnej oprogramowanie docker (wraz z narzędziem konsolowym `docker-cli`) w wersji umożliwiającej stosowanie lokalnego trybu komunikacji kontenerów docker (`--network host`)

```
> docker -v
Docker version 20.10.5, build 55c4c88
```

3. Zainstaluj w środowisku utworzonej maszyny wirtualnej aplikację klienta konsolowego serwera Redis **redis-cli** dbając o to, aby nie instalować jednocześnie serwera Redis. Skorzystaj z pakietu `redis-tools`.

```
> redis-cli -v
redis-cli 6.2.1
```

4. Uruchom predefiniowany kontener Docker serwera Redis (`redis:latest`) w trybie lokalnego trybu komunikacji kontenerów docker (`--network host`) w taki sposób, aby serwer Redis działał na porcie 6381.

```
> sudo docker run --name redisque --network host --rm redis:latest --port 6381
Password:
1:C 03 Apr 2021 08:59:19.485 # 000000000000 Redis is starting 000000000000
1:C 03 Apr 2021 08:59:19.485 # Redis version=5.0.4, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 03 Apr 2021 08:59:19.485 # Configuration loaded
1:M 03 Apr 2021 08:59:19.488 * Running mode=standalone, port=6381.
1:M 03 Apr 2021 08:59:19.488 # Server initialized
1:M 03 Apr 2021 08:59:19.488 * Ready to accept connections
```

5. Zweryfikuj działanie skonteneryzowanego serwera Redis na porcie 6381 z użyciem narzędzia klienta konsolowego serwera Redis (`redis-cli`) i polecenia `ping`.

```
> sudo docker exec -it redisque redis-cli -p 6381
127.0.0.1:6381> ping
PONG
```

6. Sprawdź w dokumentacji serwera Redis domyślny numer portu serwera i upewnij się, że serwer Redis nie działa w środowisku maszyny wirtualnej poza kontenerem Docker.
ODP. Domyślnym numerem portu jest 6379.

```
> redis-cli
Could not connect to Redis at 127.0.0.1:6379: Connection refused
not connected>
```

7. Zainstaluj wirtualne środowisko bibliotek języka programowania Python **virtualenv** z wersją 3 języka Python w katalogu o nazwie `wtivenv01`.

1. `sudo apt install virtualenv`
2. `mkdir wtivenv01`
3. `virtualenv -p python3 wtivenv01`

```
> virtualenv -p python3 wtivenv01
created virtual environment CPython3.8.2.final.0-64 in 893ms
creator CPython3macOsFramework(dest=/Users/sebastiansiejek/studia/semestr-8/wti/wtivenv01, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/sebastiansiejek/Library)
added seed packages: pip==21.0.1, setuptools==54.1.2, wheel==0.36.2
activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

8. Aktywuj środowisko wtivenv01, a następnie zainstaluj w nim bibliotekę redis.

```
~/studia/semestr-8/wti/wtiproj01
> source wtivenv01/bin/activate

~/studia/semestr-8/wti/wtiproj01
wtivenv01 > pip install redis
Collecting redis
  Using cached redis-3.5.3-py2.py3-none-any.whl (72 kB)
Installing collected packages: redis
Successfully installed redis-3.5.3
```

9. Dezaktywuj środowisko wtivenv01.
10. Zainstaluj środowisko IDE PyCharm (w wersji pycharm-community).

```
sudo snap install pycharm-community --classic
```

11. Utwórz katalog wtiproj01, a w nim plik wtiproj01_client.py z kodem aplikacji klienta serwera Redis (korzystając z biblioteki redis) tymczasowo ograniczoną do wiersza importującego bibliotekę redis oraz wiersza ustawiającego wartość dowolnie nazwanej zmiennej.

1. mkdir PycharmProjects
2. cd PycharmProjects
3. mkdir wtiproj01
4. touch wtiproj01/wtiproj01_client.py
5. pycharm-community

12. Skonfiguruj środowisko PyCharm w ten sposób, aby interpreter (tzw. jądro) Python właściwy aplikacji wtiproj01_client znajdował się w środowisku wtivenv01, a po uruchomieniu aplikacji zachowana została komunikacja z konsolą (procesem interpretera) Python.
13. Zweryfikuj poprawność konfiguracji środowiska PyCharm przez uruchomienie aplikacji wtiproj01_client, a następnie odczyt wartości uprzednio przypisanej do dowolnie nazwanej zmiennej.

```
wtivenv01 > python ./index.py
wartosc
```

14. Rozszerz implementację aplikacji klienckiej o jak największy zbiór funkcji systemu kolejkowego (funkcję tworzenia kolejki, funkcję umieszczania w kolejce komunikatu, funkcję odczytu komunikatu z kolejki, funkcję opróżniania kolejki). Aplikacja powinna demonstrować w jak najbardziej klarowny sposób użycia jak największego zbioru zaimplementowanych funkcji operacji na kolejkach.

```
import redis
import json

class Queue:

    def __init__(self):
```

```

        self.client = redis.StrictRedis(port=6381)
        self.que_name = "sample_que"

    def push(self, value):
        self.client.rpush(self.que_name, json.dumps(value))

    def list(self):
        queue_batch = self.client.lrange(self.que_name, 0, -1)
        for value_read_from_queue in queue_batch:
            print("value: ",
                  value_read_from_queue)

    def pull(self):
        values = self.client.lrange(self.que_name, 0, -1)
        self.client.ltrim(self.que_name, len(values), -1)

    def flush(self):
        self.client.flushdb()

if __name__ == "__main__":

    que = Queue()

    que.pull()

    message_to_write_to_queue_as_dict = {}
    message_to_write_to_queue_as_dict["pole1"] = "wartosc1"
    message_to_write_to_queue_as_dict["pole2"] = "wartosc2"
    que.push(message_to_write_to_queue_as_dict)

    que.list()

    que.pull()

    message_to_write_to_queue_as_dict2 = {}
    message_to_write_to_queue_as_dict2["nowe_pole"] = "wartosc1"
    que.push(message_to_write_to_queue_as_dict2)
    que.push(message_to_write_to_queue_as_dict2)

    que.list()

```

Wnioski: Udało mi się zrealizować wszystkie zadania. Zrezygnowałem jednak ze środowiska PyCharm na rzecz Visual Studio Code ponieważ pracuję w nim na co dzień i jest dla bardziej przejrzysty i wygodny,