Laboratorium z etcd

Kod (plik distributed_cache.py):

```
import json
from etcd_client import EtcdClient
class DistributedCache(object):
 def __init__(self, config_path, key_prefix, logger):
    self.etcd_client = EtcdClient(config_path, key_prefix, logger, 240) # reconnect every 4 minutes
    self.logger = logger
    self.key_prefix = key_prefix
    self.cache = {}
    self.etcd_client.add_watch_prefix_callback(self.key_prefix, self.prefix_callback)
 def prefix callback(self, event):
    print(event)
 def get(self, key: str, use_cache: bool = True) -> str:
    if use_cache and self.cache.get(key):
      return self.cache.get(key)
    full key = str(self.key_prefix) + "/" + str(key)
    data = self.etcd_client.get(full_key)[0]
    if data is not None:
       data = json.loads(data.decode("utf8"))
       self.cache[key] = data
    return data
 def put(self, key: str, value: str):
    full_key = self.key_prefix + "/" + key
    json_value = json.dumps(value)
    self.etcd_client.put(full_key, json_value)
 def delete(self, key: str):
    full_key = self.key_prefix + "/" + key
    self.etcd_client.delete(full_key)
```

Testowanie Kodu lokalnie:

Uruchomienie aplikacji

```
(venv) krzys@krzys-przystojniak:~/sem4/ibm-nosql/distributed-cache-python$ USER_KEY_PREFIX=/HardekKrzysztof/ python app.py
<etcd_client.EtcdClient object at 0x7f4a3af09a90> /HardekKrzysztof/ <bound method DistributedCache.prefix_callback of <distri
    * Serving Flask app "app" (lazy loading)
    * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
    * Debug mode: off
    * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)</pre>
```

Uruchomienie aplikacji testującej

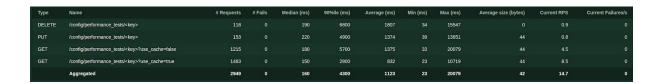
```
(venv) krzys@krzys-przystojniak:~/sem4/ibm-nosql/distributed-cache-python$ locust
[2020-05-26 23:24:03,828] krzys-przystojniak/INFO/locust.main: Starting web monitor at http://*:8089
[2020-05-26 23:24:03,828] krzys-przystojniak/INFO/locust.main: Starting Locust 0.14.6
```

Tworzę 40 użytkowników w tempie 10 na sekundę. Testy trwały około 3 minuty

Туре	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
DELETE	/config/load_tests/ <key></key>			180	340	199		525			
PUT	/config/load_tests/ <key></key>	260		180	410			716			
GET	/config/load_tests/ <key>?use_cache=false</key>	2265		160	340	183		744			
GET	/config/load_tests/ <key>?use_cache=true</key>	2545						638	44		
	Aggregated	5290		59	280	101		744	42	29.3	

Testowanie Kodu zdalnie:

Tworzę 40 użytkowników w tempie 10 na sekundę. Testy trwały około 3 minuty



Wyjaśnienie rozbieżności:

Testy pokazują, że aplikacja działa szybciej lokalnie. Jest to dosyć dziwne, ponieważ instancja na chmurze ma bezpośredni dostęp do bazy. Możliwe że był pewne opóźnienia związane z godzinami, w których wykonywałem testy

Odpowiedzi komend:

ibmcloud target

API endpoint: https://cloud.ibm.com
Region: eu-de
User: hardek@student.agh.edu.pl
Account: Krzysztof Hardek's Account (173e2df072da48edb6dc904dfaec1c84)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint: https://api.eu-de.cf.cloud.ibm.com (API version: 2.147.0)
Org: hardek-org

ibmcloud cf app distributed-cache

ibmcloud cf logs distributed-cache --recent | head -n 20

```
Provide price of the price of t
```