Cloud computing Lab3 – sprawozdanie

# Rozwiązanie Etap I – cz. I

## Treść

A screenshot of a computer

Description automatically generated

## Tworzenie bazy danych

A screenshot of a computer

Description automatically generated

## Połączenie z bazą danych

### Kod

import psycopg2  
  
  
def connect\_to\_postgres(host, port, database, user, password):  
  
 try:  
 connection = psycopg2.connect(  
 host=host,  
 port=port,  
 database=database,  
 user=user,  
 password=password  
 )  
 print("Connected")  
 return connection  
 except Exception as e:  
 print("Error connecting")  
 print(e)  
 return None  
  
  
def main():  
 host = "database-1-user15.cj888w4mq49y.us-west-2.rds.amazonaws.com"  
 port = 5432  
 database = "postgres"  
 user = "postgres"  
 password = "radziu13"  
  
 conn = connect\_to\_postgres(host, port, database, user, password)  
  
 if conn:  
 try:  
 cursor = conn.cursor()  
  
 cursor.execute("SELECT version();")  
 version = cursor.fetchone()  
 print(f"{version[0]}")  
  
 finally:  
 conn.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

### Wykonanie

*Connected*

*PostgreSQL 16.3 on x86\_64-pc-linux-gnu, compiled by gcc (GCC) 7.3.1 20180712 (Red Hat 7.3.1-17), 64-bit*

Process finished with exit code 0

## Dodawanie I pobieranie danych

### Kod

import psycopg2  
  
  
def connect\_to\_postgres(host, port, database, user, password):  
  
 try:  
 connection = psycopg2.connect(  
 host=host,  
 port=port,  
 database=database,  
 user=user,  
 password=password  
 )  
 print("Connected")  
 return connection  
 except Exception as e:  
 print("Error connecting")  
 print(e)  
 return None  
  
  
def create\_table\_query():  
 return """CREATE TABLE IF NOT EXISTS person (  
 first\_name VARCHAR(255) NOT NULL,  
 last\_name VARCHAR(255) NOT NULL,  
 age INT NOT NULL,  
 email VARCHAR(255) NOT NULL UNIQUE  
);"""  
  
def pupulate\_table\_query():  
 return """INSERT INTO person (first\_name, last\_name, age, email)   
VALUES   
 ('John', 'Doe', 30, 'john.doe@example.com'),  
 ('Jane', 'Smith', 25, 'jane.smith@example.com'),  
 ('Michael', 'Johnson', 40, 'michael.johnson@example.com'),  
 ('Emily', 'Davis', 22, 'emily.davis@example.com'),  
 ('William', 'Brown', 35, 'william.brown@example.com')  
ON CONFLICT DO NOTHING;"""  
  
def get\_data\_from\_table\_query():  
 return """SELECT \* FROM person;"""  
  
  
def main():  
 host = "database-1-user15.cj888w4mq49y.us-west-2.rds.amazonaws.com"  
 port = 5432  
 database = "postgres"  
 user = "postgres"  
 password = "radziu13"  
  
 conn = connect\_to\_postgres(host, port, database, user, password)  
  
 if conn:  
 try:  
 cursor = conn.cursor()  
  
 cursor.execute("SELECT version();")  
 version = cursor.fetchone()  
 print(f"{version[0]}")  
  
 cursor.execute(create\_table\_query())  
 conn.commit()  
  
 cursor.execute(pupulate\_table\_query())  
 conn.commit()  
  
 cursor.execute(get\_data\_from\_table\_query())  
 rows = cursor.fetchall()  
 print("Person data:")  
 for row in rows:  
 print(row)  
  
 finally:  
 conn.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

### Wykonanie

*Connected*

*PostgreSQL 16.3 on x86\_64-pc-linux-gnu, compiled by gcc (GCC) 7.3.1 20180712 (Red Hat 7.3.1-17), 64-bit*

*Person data:*

*('John', 'Doe', 30, 'john.doe@example.com')*

*('Jane', 'Smith', 25, 'jane.smith@example.com')*

*('Michael', 'Johnson', 40, 'michael.johnson@example.com')*

*('Emily', 'Davis', 22, 'emily.davis@example.com')*

*('William', 'Brown', 35, 'william.brown@example.com')*

*Process finished with exit code 0*

# Rozwiązanie Etap I – cz. II

A screenshot of a computer

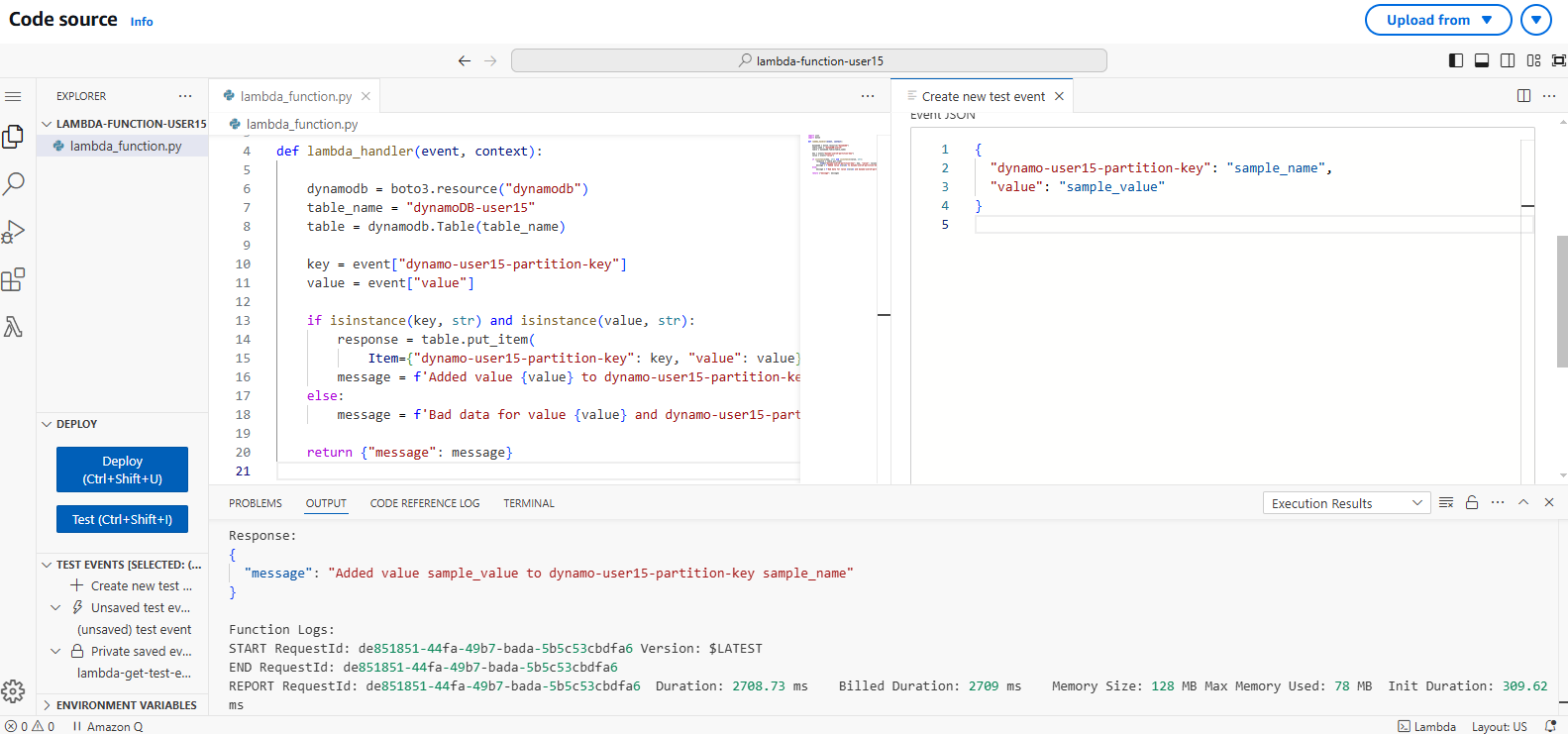
Description automatically generated

## Tworzenie tabeli

A screenshot of a computer

Description automatically generated

## Tworzenie funkcji lambda do dodawania rekordu z walidacją + test



## Tworzenie funkcji lambda do zwrócenia rekordów na podstawie parametrów + test

A screenshot of a computer

Description automatically generated

# Rozwiązanie Etap II

A screenshot of a computer

Description automatically generated

## Tworzenie API Gateway wywołujący funkcję lambda

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Wywołanie endpointa API Gateway z lokalnej aplikacji

### Kod

import json  
  
import requests  
  
URL = 'https://9ya1ocqd37.execute-api.us-west-2.amazonaws.com/stage-user15'  
  
  
def add\_record(key, value):  
  
 url = URL + '/resouce-put-user15'  
  
 payload = {  
 "dynamo-user15-partition-key": key,  
 "value": "value"  
 }  
  
 headers = {'content-type': 'application/json'}  
  
 try:  
 response = requests.post(url, data=json.dumps(payload), headers=headers)  
 response.raise\_for\_status()  
 return response.json()  
 except requests.exceptions.HTTPError as e:  
 print('Http Error')  
 print(e)  
  
 return None  
  
  
def get\_record(key):  
  
 url = URL + '/resouce-get-user15'  
  
 payload = {"dynamo-user15-partition-key": key}  
  
 headers = {'content-type': 'application/json'}  
  
 try:  
 response = requests.post(url, data=json.dumps(payload), headers=headers)  
 response.raise\_for\_status()  
 return response.json()  
 except requests.exceptions.HTTPError as e:  
 print('Http Error')  
 print(e)  
  
 return None  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 key = 'local\_sample\_key'  
 value = 'local\_sample\_value'  
  
 print(add\_record(key, value))  
 print(get\_record(key))

### Wykonanie

*{'message': 'Added value value to dynamo-user15-partition-key local\_sample\_key'}*

*{'message': 'Got record from key local\_sample\_key value: value'}*

*Process finished with exit code 0*

## Tworzenie bucketu w S3

### Tworzenie bucketu

A screenshot of a computer

Description automatically generated

### Wgrywanie pliku

A screenshot of a computer

Description automatically generated

### Funkcje do wyświetlania listy plików i wyszukania pliku po nazwie

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Rozwiązanie Etap III

A white text with black text

Description automatically generated

## Tworzenie maszyny wirtualnej

A screenshot of a computer

Description automatically generated

## Nawiązanie połączenia z maszyną wirtualną

A screenshot of a computer

Description automatically generated

## Nawiązanie połączenia z bazą danych, dodanie/pobranie rekordu

A screenshot of a computer

Description automatically generated