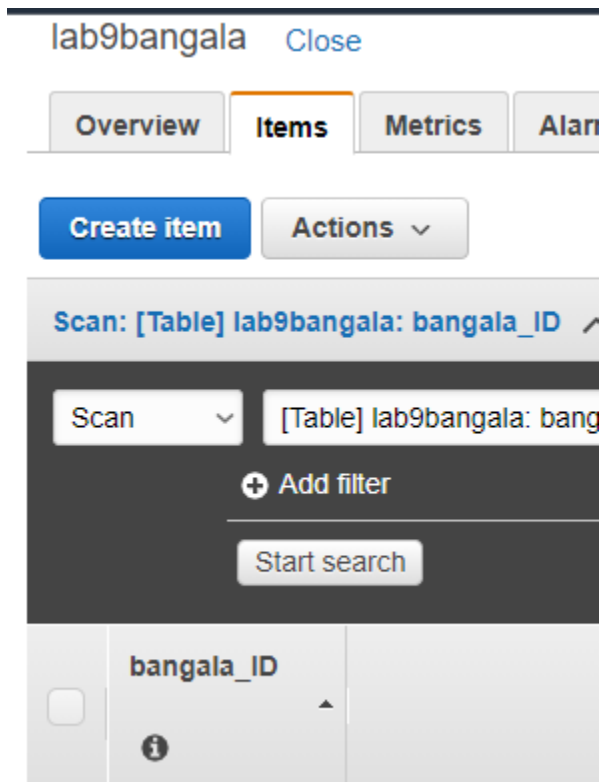


Zadania lab 9 Przemysław Kawa

1. Zaprojektuj własną strukturę tabeli w bazie DynamoDB, do które dane zostaną wprowadzone w postaci wywołanie AWS API Gateway. Przedstaw kilka przykładów komunikacji z usługą, która zapisze informacje w utworzonej bazie danych.



REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:
Lambda, HTTP, AWS Services

Import

Build



Amazon API Gateway

APIs > Create

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API ☐ Import from Swagger or Open API 3 ☐ Example API

Settings

Choose a friendly name and description for your API.

API name*

lab9

Description

Kawa robot

Endpoint Type

Regional



* Required

APIs

Custom Domain Names

VPC Links

API: lab9

Resources

Stages

Authorizers

Resources

Actions



Methods

RESOURCE ACTIONS

Create Method

Create Resource

Enable CORS

Edit Resource Documentation

API ACTIONS

Deploy API

Import API

Edit API Documentation

Delete API

Create function [Info](#)

Choose one of the following options to create your function

Author from scratch ☒

Start with a simple Hello World example.

Basic information

Function name

Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no space

Runtime [Info](#)

Choose the language to use to write your function.

Permissions [Info](#)

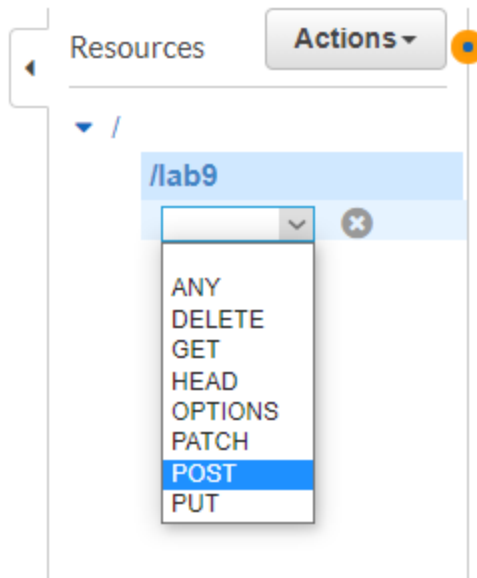
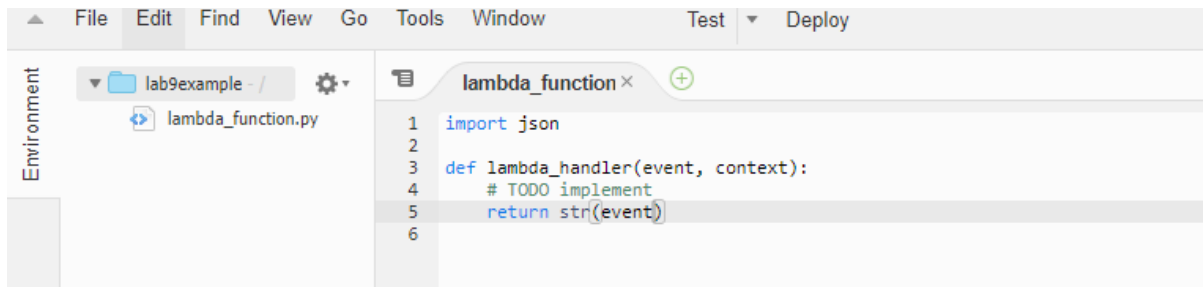
By default, Lambda will create an execution role with permissions

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To cr

- ☐ Create a new role with basic Lambda permissions
- ☒ Use an existing role
- ☐ Create a new role from AWS policy templates



/lab9 - POST - Setup

Choose the integration point for your new method.

Integration type ☒ Lambda Function ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

Use Lambda Proxy integration ☐ ⓘ

Lambda Region

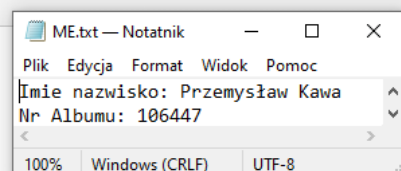
Lambda Function ⓘ

Use Default Timeout ☒ ⓘ

Save

Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:us-east-1:138084980886:function:lab9example



Cancel

OK

TEST TIME:

make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

{lab9}

param1=value1¶m2=value2

Headers

{lab9}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg.
Accept:application/json.

Stage Variables

No [stage variables](#) exist for this method.

Request Body

```
1 {  
2   "bangala_ID": "1234",  
3   "Imie": "Przemysław", |  
4   "Nazwisko": "Kawa"  
5 }
```

Status: 200

Jest Dobrze

Latency: 403 ms

Response Body

```
"{'bangala_ID': '1234', 'Imie': 'Przemysław', 'Nazwisko': 'Kawa'}"
```

Response Headers

```
{ "X-Amzn-Trace-Id": "Root=1-5ff881cc-08e9396a9317cd0e569d2925;Sampled=0", "Content-Type": "application/json" }
```

Logs

```
Execution log for request df0b4a83-2042-43ee-b283-da068c8a2bb1
Fri Jan 08 16:01:16 UTC 2021 : Starting execution for request: df0b4a83-2042-43ee-b283-da068c8a2bb1
Fri Jan 08 16:01:16 UTC 2021 : HTTP Method: POST, Resource Path: /lab9
Fri Jan 08 16:01:16 UTC 2021 : Method request path: {}
Fri Jan 08 16:01:16 UTC 2021 : Method request query string: {}
Fri Jan 08 16:01:16 UTC 2021 : Method request headers: {}
Fri Jan 08 16:01:16 UTC 2021 : Method request body before transformations: {
    "bangala_ID": "1234",
    "Imie": "Przemysław",
    "Nazwisko": "Kawa"
}
Fri Jan 08 16:01:16 UTC 2021 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:138084980886:function:lab9example/invocations
Fri Jan 08 16:01:16 UTC 2021 : Endpoint request headers: {x-amzn-lambda-integration-tag=df0b4a83-2042-43ee-b283-da068c8a2bb1, Authorization=*****
*****
*****
*****
*****
```

```
1 import json
2 import boto3
3 dynamodb = boto3.resource('dynamodb')
4 table = dynamodb.Table('lab9bangala')
5 def lambda_handler(event, context):
6     # TODO implement
7     table.put_item(Item=event)
8     return [{"code": "200", "message": "Job Done!"}]
9
```

Make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

{lab9}

param1=value1¶m2=value2

Headers

{lab9}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables

No [stage variables](#) exist for this method.

Request Body

```
1 {
2   "bangala_ID": "1234",
3   "Imie": "NiePrzemek",
4   "Nazwisko": "NieKawa"
5 }
```

Request: /lab9

Status: 200

Latency: 832 ms

Response Body

```
{
  "code": 200,
  "message": "Job Done!"
}
```

Response Headers

```
{"X-Amzn-Trace-Id": "Root=1-5ff88583-d29a15bfa83d8e30f55d381e;Sampled=0", "Content-Type": "application/json"}
```

Logs

```
Execution log for request 4b55a1cc-4ab3-4fe8-b8c8-7aae324f609d
Fri Jan 08 16:17:07 UTC 2021 : Starting execution for request: 4b55a1cc-4ab3-4fe8-b8c8-7aae324f609d
Fri Jan 08 16:17:07 UTC 2021 : HTTP Method: POST, Resource Path: /lab9
Fri Jan 08 16:17:07 UTC 2021 : Method request path: {}
Fri Jan 08 16:17:07 UTC 2021 : Method request query string: {}
Fri Jan 08 16:17:07 UTC 2021 : Method request headers: {}
Fri Jan 08 16:17:07 UTC 2021 : Method request body before transformations: {
  "bangala_ID": "1234",
  "Imie": "NiePrzemek",
  "Nazwisko": "NieKawa"
}
```

lab9bangala Close

Overview

Items

Metrics

Alarms

Capacity

Create item

Actions

Scan: [Table] lab9bangala: bangala_ID

Scan

[Table] lab9bangala: bangala_ID

+ Add filter

Start search

bangala_ID

Imie

Nazwisko

1234

NiePrzeme

NieKawa

“Przedstaw kilka przykładów komunikacji z usługą, która zapisze informacje w utworzonej bazie danych.” Przykładów nie sposobów... ok Jedziemy!

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

{lab9}

param1=value1¶m2=value2

Headers

{lab9}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables

No [stage variables](#) exist for this method.

Request Body

```
1 {
2   "bangala_ID": "123",
3   "Imie": "NiePrzemek",
4   "Nazwisko": "NieKawa"
5 }
```

Make a test call to your method with the provided input

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

Query Strings

{lab9}

param1=value1¶m2=value2

Headers

{lab9}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables

No [stage variables](#) exist for this method.

Request Body

```
1 {
2   "bangala_ID": "13",
3   "Imie": "BycMozeBaszcz",
4   "Nazwisko": "NOTE NieRobSprawozdanJakJestesGlodny"
5 }
```

Request: /lab9

Status: 200

Latency: 830 ms

Response Body

```
{
  "code": 200,
  "message": "Job Done!"
}
```

Response Headers

```
{"X-Amzn-Trace-Id": "Root=1-5ff88748-6feaf236e44fb64c8077279c;Sampled=0", "Content-Type": "application/json"}
```

Logs

```
Execution log for request bd711482-248b-4d66-abac-6e17ac36f90f
Fri Jan 08 16:24:40 UTC 2021 : Starting execution for request: bd711482-248b-4d66-abac-6e17ac36f90f
Fri Jan 08 16:24:40 UTC 2021 : HTTP Method: POST, Resource Path: /lab9
Fri Jan 08 16:24:40 UTC 2021 : Method request path: {}
Fri Jan 08 16:24:40 UTC 2021 : Method request query string: {}
Fri Jan 08 16:24:40 UTC 2021 : Method request headers: {}
Fri Jan 08 16:24:40 UTC 2021 : Method request body before transformations: {
  "bangala_ID": "123",
  "Imie": "NiePrzemek",
  "Nazwisko": "NieKawa"
}
```

Request: /lab9

Status: 200

Latency: 242 ms

Response Body

```
{
  "code": 200,
  "message": "Job Done!"
}
```

Response Headers

```
{"X-Amzn-Trace-Id": "Root=1-5ff88797-17caa09501cb0eda9fab96e6;Sampled=0", "Content-Type": "application/json"}
```

Logs

```
Execution log for request 91227724-e53c-47af-9f10-1efc0e276c7f
Fri Jan 08 16:25:59 UTC 2021 : Starting execution for request: 91227724-e53c-47af-9f10-1efc0e276c7f
Fri Jan 08 16:25:59 UTC 2021 : HTTP Method: POST, Resource Path: /lab9
Fri Jan 08 16:25:59 UTC 2021 : Method request path: {}
Fri Jan 08 16:25:59 UTC 2021 : Method request query string: {}
Fri Jan 08 16:25:59 UTC 2021 : Method request headers: {}
Fri Jan 08 16:25:59 UTC 2021 : Method request body before transformations: {
  "bangala_ID": "13",
  "Imie": "BycMozeBaszcz",
  "Nazwisko": "NOTE NieRobSprawozdanJakJestesGlodny"
}
```

2. Przetwórz proces konfiguracji i uruchamiania środowiska Jupyter Notebook w chmurze publicznej AWS. Na bazie dołączonych przykładów, napisz kilka skryptów w dowolnej dziedzinie (np. analiza danych), które przedstawiają wyniki w postaci graficznej (np. tabela, wykres).

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-00ddb0e5626798373

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebs Virtualization type: hvm

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security group name	launch-wizard-2
Description	launch-wizard-2 created 2021-01-08T17:44:59.388+01:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	0.0.0.0/0	

i-0522f78d10b82cb87

Running

🔍

t2.micro

✔️ 2/2 checks ...

⌵ 1/2 has no data

sg-0867634ef33cd9e7 (Jupyter)

Inbound rules

Filter rules

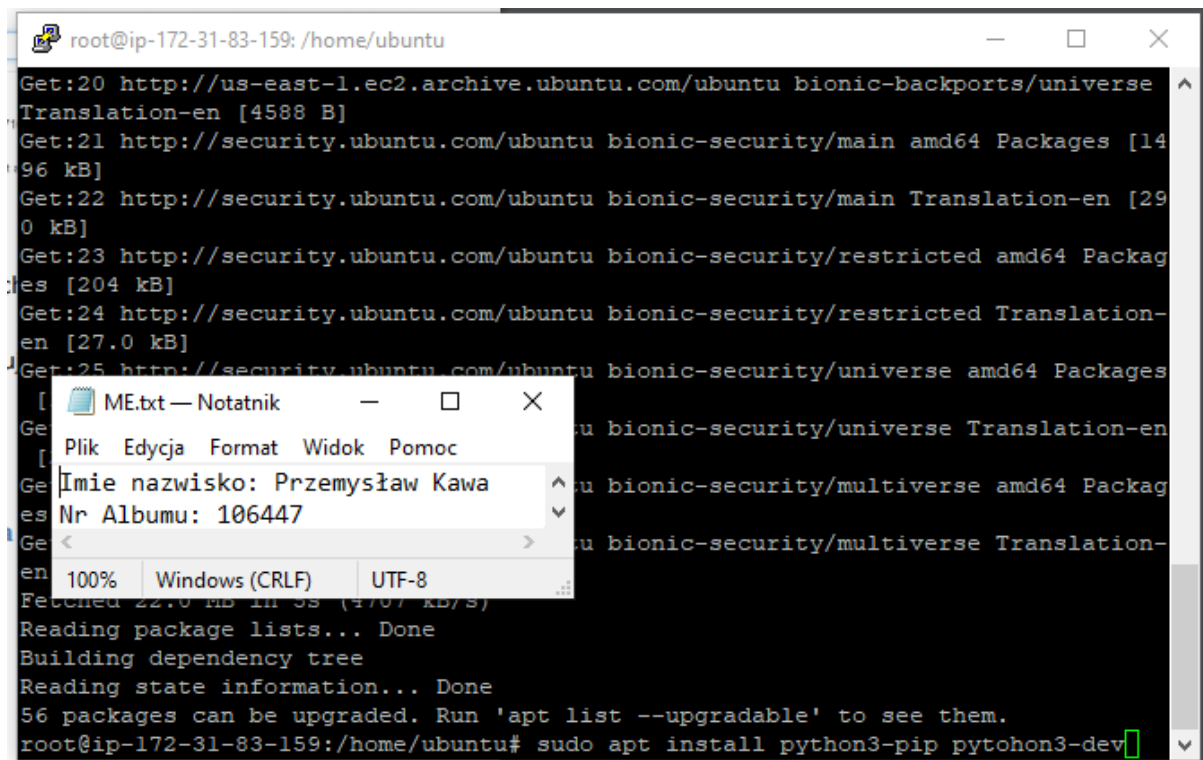
Port range	Protocol	Source	Security groups	launch-wizard-2	Jupyter
22	TCP	0.0.0.0/0	launch-wizard-2, Jupyter	✔️	✔️
8888	TCP	0.0.0.0/0	Jupyter		✔️
8888	TCP	::/0	Jupyter		✔️
22	TCP	::/0	Jupyter		✔️
443	TCP	0.0.0.0/0	Jupyter		✔️

Instance ID

Public IPv4 address

📄 i-0522f78d10b82cb87

📄 52.90.88.7 | open address 🔗



The image shows a terminal window with the following output:

```
root@ip-172-31-83-159: /home/ubuntu
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1496 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [290 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [204 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [27.0 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1496 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [4588 B]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1496 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [4588 B]
Fetched 22.0 MB in 3s (4707 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
56 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-83-159:/home/ubuntu# sudo apt install python3-pip python3-dev
```

An overlaid Notepad window titled "ME.txt — Notatnik" contains the following text:

Imię nazwisko: Przemysław Kawa
Nr Albumu: 106447

Zastosowane polecenia:

```
sudo apt update
```

```
sudo apt install python3-pip python3-dev
```

Przygotowanie środowiska Python Virtual Environment dla Jupyter

```
sudo -H pip3 install --upgrade pip
```

```
sudo -H pip3 install virtualenv
```

Koniec roota do : exit

```
mkdir ~/my_project_dir
```

```
cd ~/my_project_dir
```

```
virtualenv my_project_env
```

```
source my_project_env/bin/activate
```

```
ubuntu@ip-172-31-83-159: ~/my_project_dir
Downloading typing_extensions-3.7.4.3-py3-none-any.whl (22 kB)
Collecting zipp>=0.5
12 Downloading zipp-3.4.0-py3-none-any.whl (5.2 kB)
Installing collected packages: zipp, typing-extensions, importlib-resources, imp
ortlib-metadata, filelock, distlib, appdirs, virtualenv
Successfully installed appdirs-1.4.4 distlib-0.3.1 filelock-3.0.12 importlib-met
adata-3.3.0 importlib-resources-4.1.1 typing-extensions-3.7.4.3 virtualenv-20.2.
2 zipp-3.4.0
root@ip-172-31-83-159:/home/ubuntu# exit
exit
ubuntu@ip-172-31-83-159:~$ mkdir ~/my_project_dir
ubuntu@ip-172-31-83-159:~$
ubuntu@ip-172-31-83-159:~$ cd ~/my_project_dir
ubuntu@ip-172-31-83-159:~/my_project_dir$ virtualenv my_project_env
created virtual environment CPython3.6.9.final.0-64 in 818ms
  creator CPython3Posix(dest=/home/ubuntu/my_project_dir/my_project_env, clear=F
alse, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle
, via=copy, app_data_dir=/home/ubuntu/.local/share/virtualenv)
  added seed packages: pip==20.3.1, setuptools==51.0.0, wheel==0.36.1
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,Pyt
honActivator,XonshActivator
ubuntu@ip-172-31-83-159:~/my_project_dir$ source my_project_env/bin/activate
(my project env) ubuntu@ip-172-31-83-159:~/my_project_dir$
```

ME.txt — Notatnik

Plik Edycja Format Widok Pomoc

Imię nazwisko: Przemysław Kawa

Nr Albumu: 106447

< >

100% Windows (CRLF) UTF-8

```
ubuntu@ip-172-31-83-159: ~/my_project_dir
[I 17:06:35.017 NotebookApp] Writing notebook server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/notebook_cookie_secret
[I 17:06:35.329 NotebookApp] Serving notebooks from local directory: /home/ubuntu/my_project_dir
[I 17:06:35.329 NotebookApp] Jupyter Notebook 6.1.6 is running at:
[I 17:06:35.329 NotebookApp] http://localhost:8888/?token=8504694f94cdff542d316a6ee2cbd26befdfa489a37c8ced
[I 17:06:35.329 NotebookApp] or http://127.0.0.1:8888/?token=8504694f94cdff542d316a6ee2cbd26befdfa489a37c8ced
[I 17:06:35.330 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 17:06:35.334 NotebookApp] No web browser found: could not locate runnable browser.
[C 17:06:35.334 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-18703-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=8504694f94cdff542d316a6ee2cbd26befdfa489a37c8ced
or http://127.0.0.1:8888/?token=8504694f94cdff542d316a6ee2cbd26befdfa489a37c8ced
```

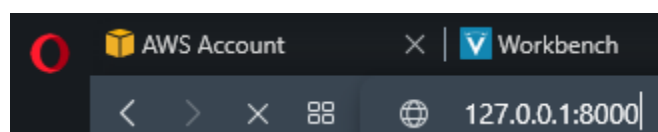
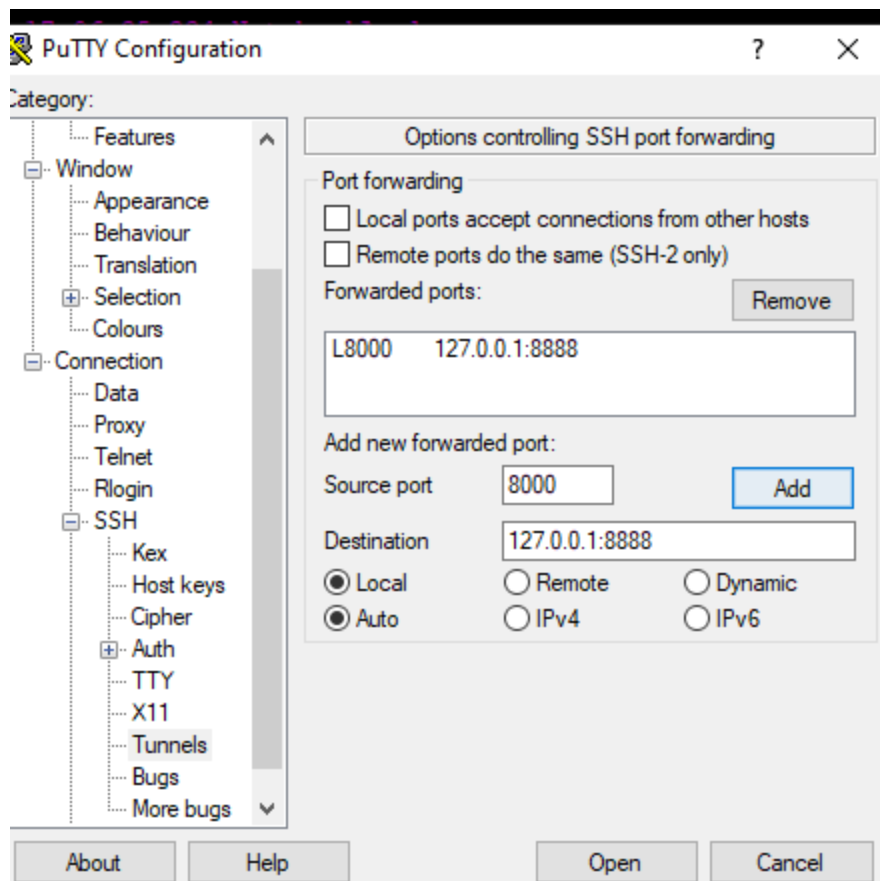
ME.txt — Notatnik

Plik Edycja Format Widok Pomoc

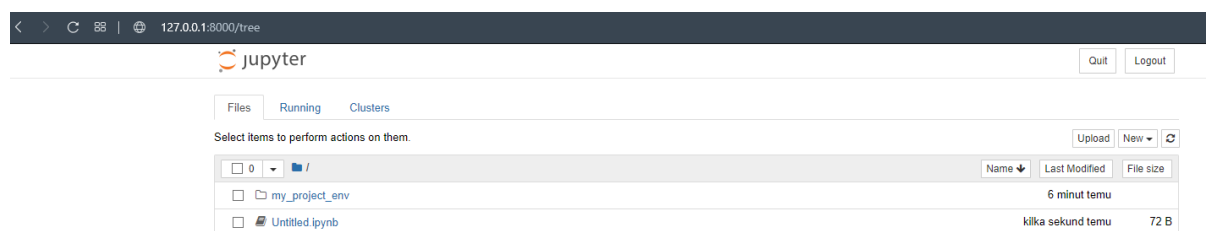
Imię nazwisko: Przemysław Kawa

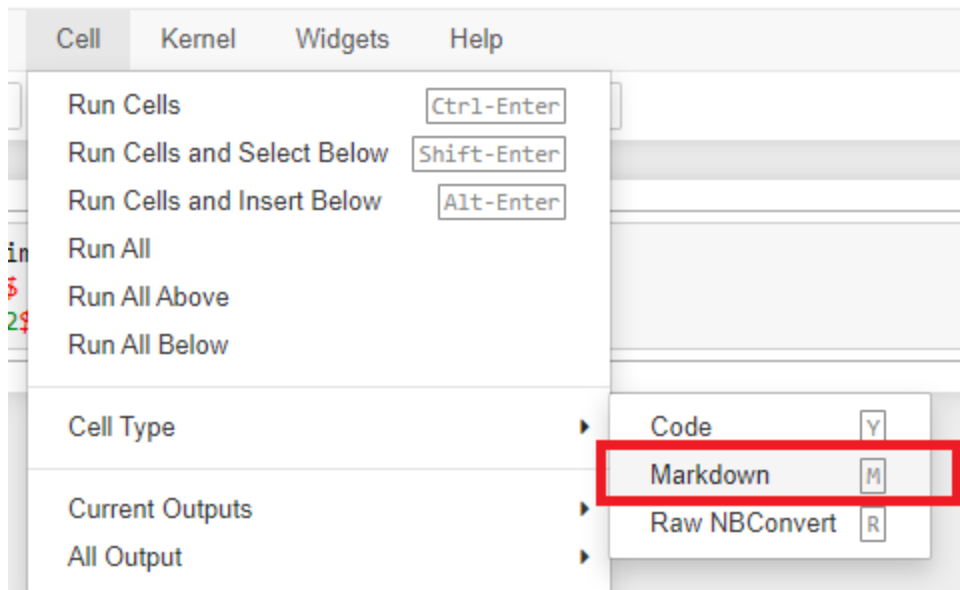
Nr Albumu: 106447

100% Windows (CRLF) UTF-8



```
or http://127.0.0.1:8888/?token=8504694f94cdf542d316a6ee2cbd26befdfa489a37c8ced
```





```
In [1]: x = 2
        y = x**2
        print(y)
```

4

Na bazie dołączonych przykładów, napisz kilka skryptów z dowolnej dziedziny (np. analiza danych), które przedstawiają wyniki w postaci graficznej (np. tabela, wykres).

```
(my_project_env) ubuntu@ip-172-31-83-159:~/my_project_dir$ pip install watermark
```

```
In [1]: %load_ext watermark
        %watermark -a 'Sebastian Raschka' -u -d -v
```

```
Sebastian Raschka
last updated: 2021-01-08

CPython 3.6.9
IPython 7.16.1
```

```
In [2]: def fibo_recurse(n):
        if n <= 1:
            return n
        else:
            return fibo_recurse(n-1) + fibo_recurse(n-2)
```

```
print(fibo_recurse(0))
print(fibo_recurse(1))
print(fibo_recurse(10))
```

```
0
1
55
```

```
In [3]: def fibo_dynamic(n):
        f, f_minus_1 = 0, 1
        for i in range(n):
            f_minus_1, f = f, f + f_minus_1
        return f
```

```
print(fibo_dynamic(0))
print(fibo_dynamic(1))
print(fibo_dynamic(10))
```

```
0
1
55
```

```
In [*]: import timeit

funcs = ['fibonacci_recurse', 'fibonacci_dynamic']
orders_n = list(range(0, 50, 10))
times_n = {f:[] for f in funcs}

for n in orders_n:
    for f in funcs:
        times_n[f].append(min(timeit.Timer('%s(n)' % f,
            'from __main__ import %s, n' % f)
            .repeat(repeat=3, number=5)))
```

```
In [*]: %matplotlib inline
import matplotlib.pyplot as plt

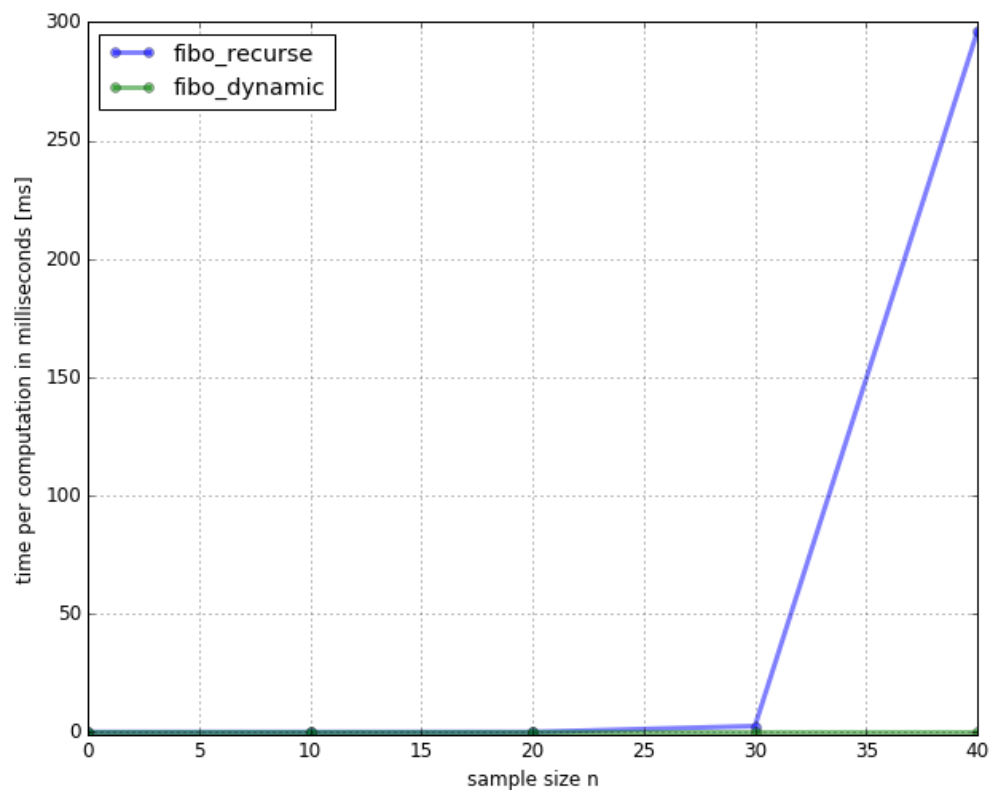
def plot_timing():

    labels = [('fibonacci_recurse', 'fibonacci_recurse'),
              ('fibonacci_dynamic', 'fibonacci_dynamic')]

    plt.rcParams.update({'font.size': 12})

    fig = plt.figure(figsize=(10, 8))
    for lb in labels:
        plt.plot(orders_n, times_n[lb[0]],
            alpha=0.5, label=lb[1], marker='o', lw=3)
    plt.xlabel('sample size n')
    plt.ylabel('time per computation in milliseconds [ms]')
    plt.legend(loc=2)
    plt.ylim([-1, 300])
    plt.grid()
    plt.show()
```

```
In [*]: plot_timing();
```



Wnioski:

- API GATEWAY było by bardzo przydatne do wykorzystania w serwisach webowych korzystających z S3 jako magazyn danych (łatwiejszy dostęp). Ale potrzebne są dodatkowe lekcje do tego poziomu.