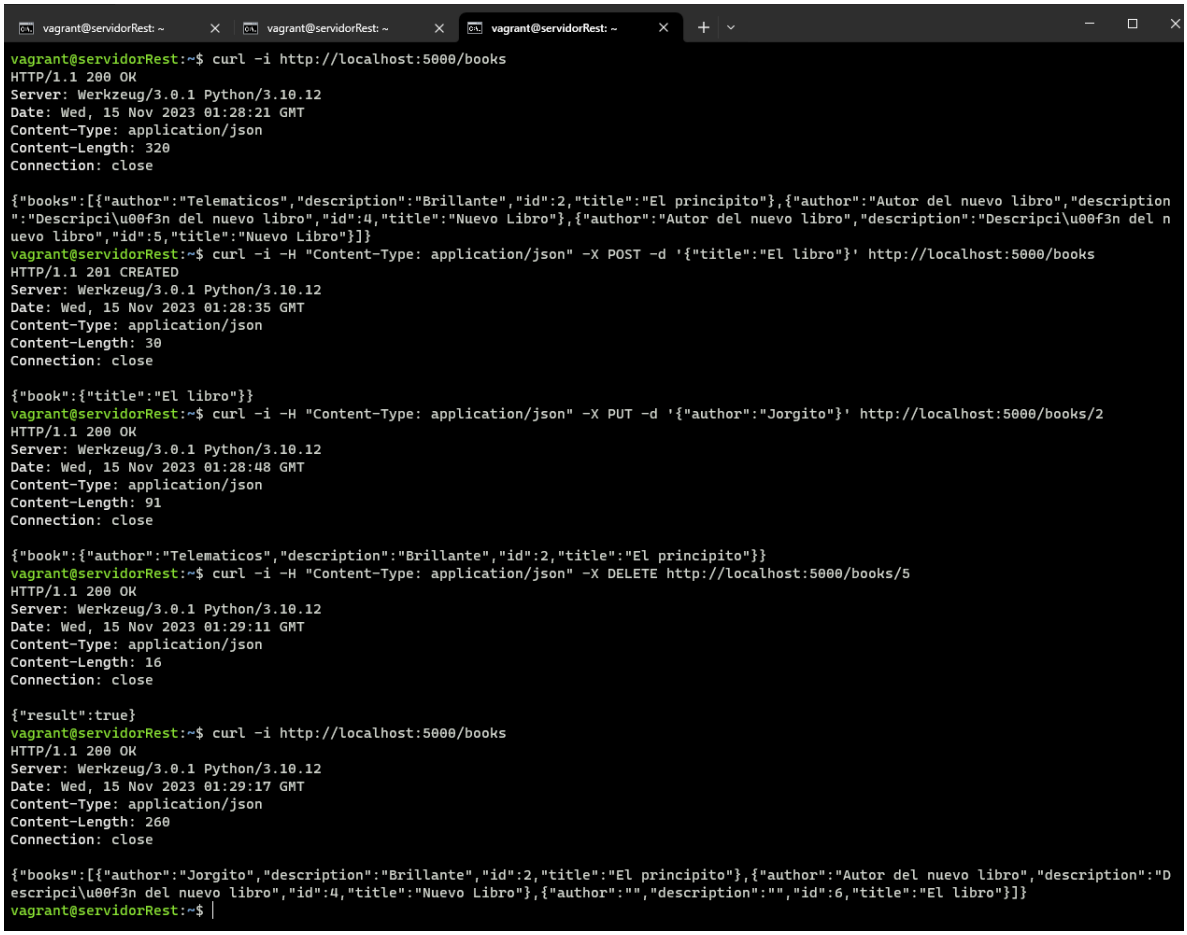
Paso a Paso punto 1:

Instalación y configuración de Prometheus:

1. Realizamos lo descrito en la guía del parcial y lo comprobamos:



2. Codificamos el codifo que funciona como cliente para la API REST probada Enel anterior punto con todas las funciones:

```
import requests
import json

base_url = 'http://localhost:5000/books'

# Función para realizar una solicitud GET
def get_books():
    response = requests.get(base_url)
    print(response.json())

# Función para realizar una solicitud GET por ID
def get_book(book_id):
    url = f'{base_url}/{book_id}'
    response = requests.get(url)
    print(response.json())
```

```
# Función para realizar una solicitud POST
def create_book(title, description, author):
    headers = {'Content-Type': 'application/json'}
    data = {'title': title, 'description': description, 'author':
author}
    response = requests.post(base_url, headers=headers,
data=json.dumps(data))
    print(response.json())

# Función para realizar una solicitud PUT
def update_book(book_id, author):
    url = f'{base_url}/{book_id}'
    headers = {'Content-Type': 'application/json'}
    data = {'author': author}
    response = requests.put(url, headers=headers,
data=json.dumps(data))
    print(response.json())

# Función para realizar una solicitud DELETE
def delete_book(book_id):
    url = f'{base_url}/{book_id}'
    response = requests.delete(url)
    print(response.json())

# Pruebas- Se descomenta la funcion que se desea usar

# Obtener todos los libros
get_books()

# Obtener un libro por ID
#get_book(1)

# Agregar un nuevo libro
#create_book('Nuevo Libro', 'Descripción del nuevo libro', 'Autor
del nuevo libro')

# Actualizar un libro por ID
#update_book(2, 'Telematicos')
```

- Para comprobar su funcionamiento, usaremos todas las solicitudes, usamos el código:

```
python3 cliente.py
```