



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

---

Институт информационных технологий (ИИТ)

Кафедра практической и прикладной информатики (ППИ)

## **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Микросервисная архитектура»

### **Практическое задание № 5**

Студент группы *ИКБО-25-20, Валиков Кирилл*  
*Юрьевич*

---

подпись)

Преподаватель *Лантев И.А.*

---

подпись)

Отчет представлен \_\_\_\_\_ 202\_\_ г.

# 1 Создать автоудаляемую очередь.

Брокер сообщений, запущенный в контейнере (рисунок 1)

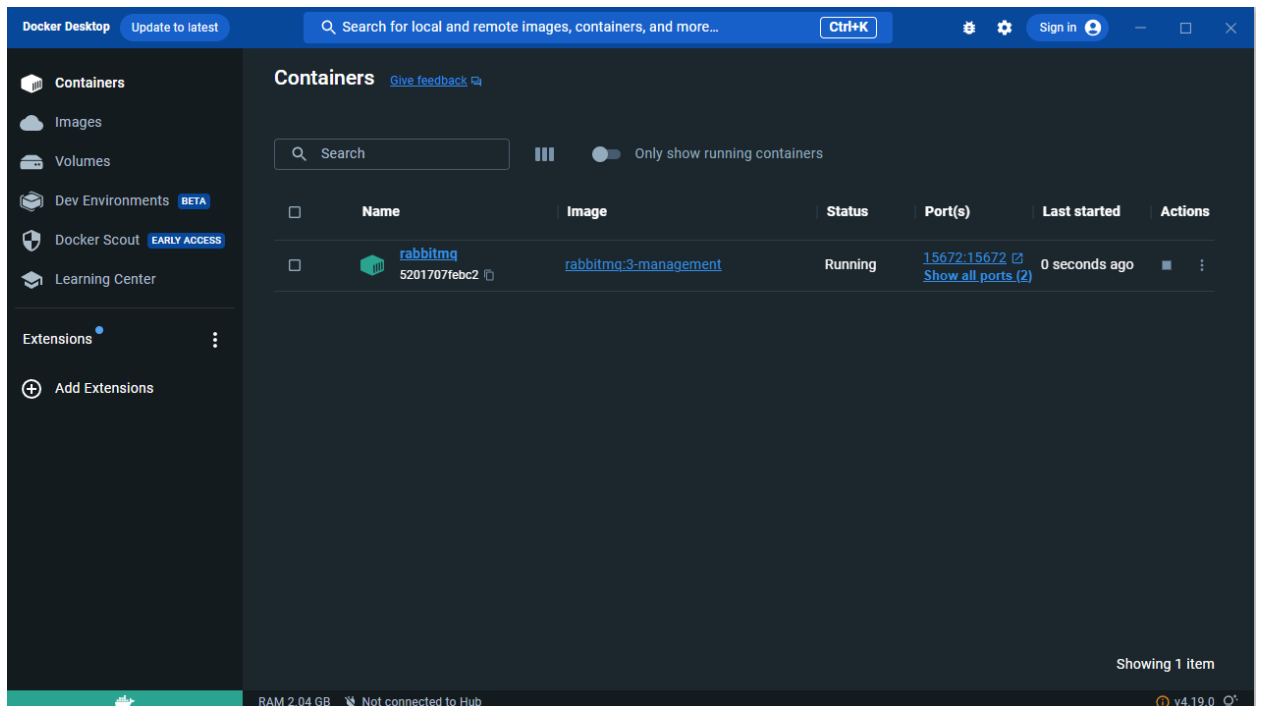


Рисунок 1. Брокер сообщений, запущенный в контейнере

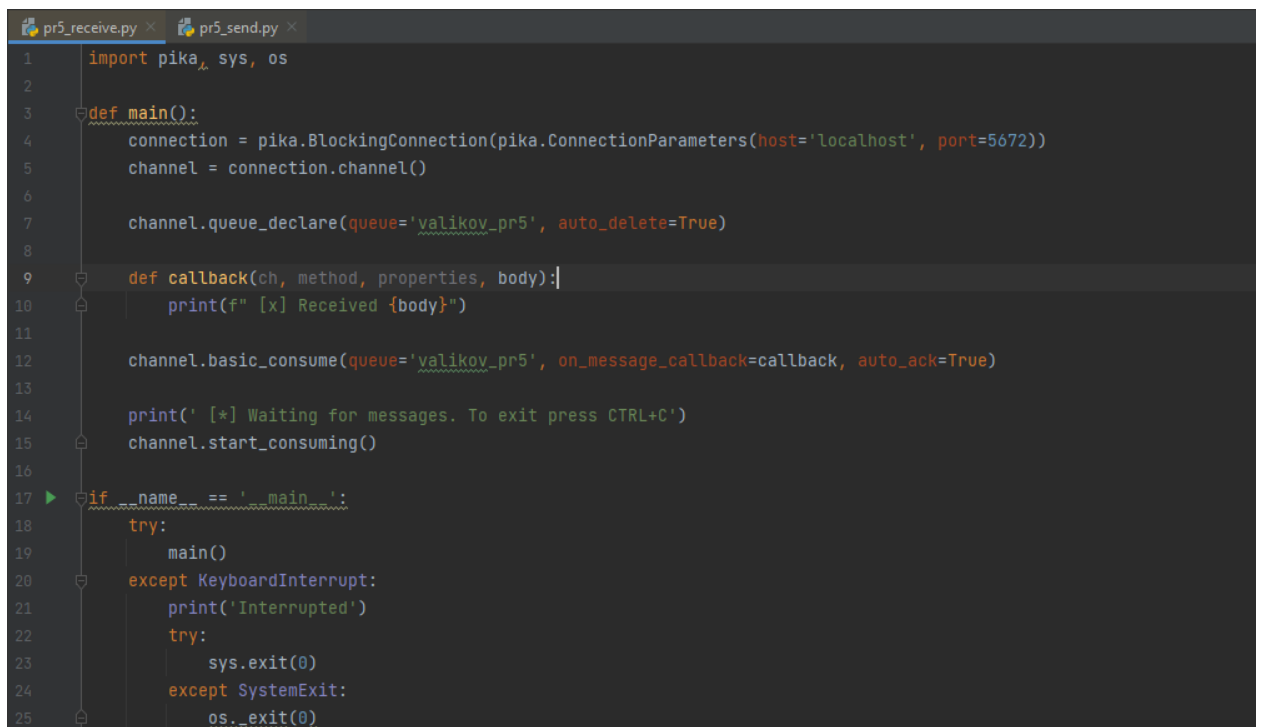


Рисунок 2. pr5\_receive.py

```
pr5_receive.py x pr5_send.py x
1 import pika
2
3 connection = pika.BlockingConnection(pika.ConnectionParameters(host='localhost', port=5672))
4 channel = connection.channel()
5
6 channel.queue_declare(queue='valikov_pr5', auto_delete=True)
7
8 channel.basic_publish(exchange='',
9                       routing_key='valikov_pr5',
10                      body='Hello World!')
11 print("[x] Sent 'Hello World!'")
12
13 connection.close()
```

Рисунок 3. pr5\_send.py

Queues

▼ All queues (1)

Pagination

Page 1 ▼ of 1 - Filter:  ☐ Regex ?

Overview					Messages			Message rates			+/-
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
/	valikov_pr5	classic	AD	running	0	0	0				

► Add a new queue

Рисунок 4. Созданная очередь, после запуска pr5\_receive.py

```
Windows PowerShell
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\pr5_receive.py
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
[x] Received b'Hello World!'
[x] Received b'Hello World!'

Windows PowerShell
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\pr5_send.py
[x] Sent 'Hello World!'
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\pr5_send.py
[x] Sent 'Hello World!'
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\pr5_send.py
[x] Sent 'Hello World!'
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\pr5_send.py
[x] Sent 'Hello World!'
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> _
```

Рисунок 5. Проверка работы

Queues

▼ All queues (0)

Pagination

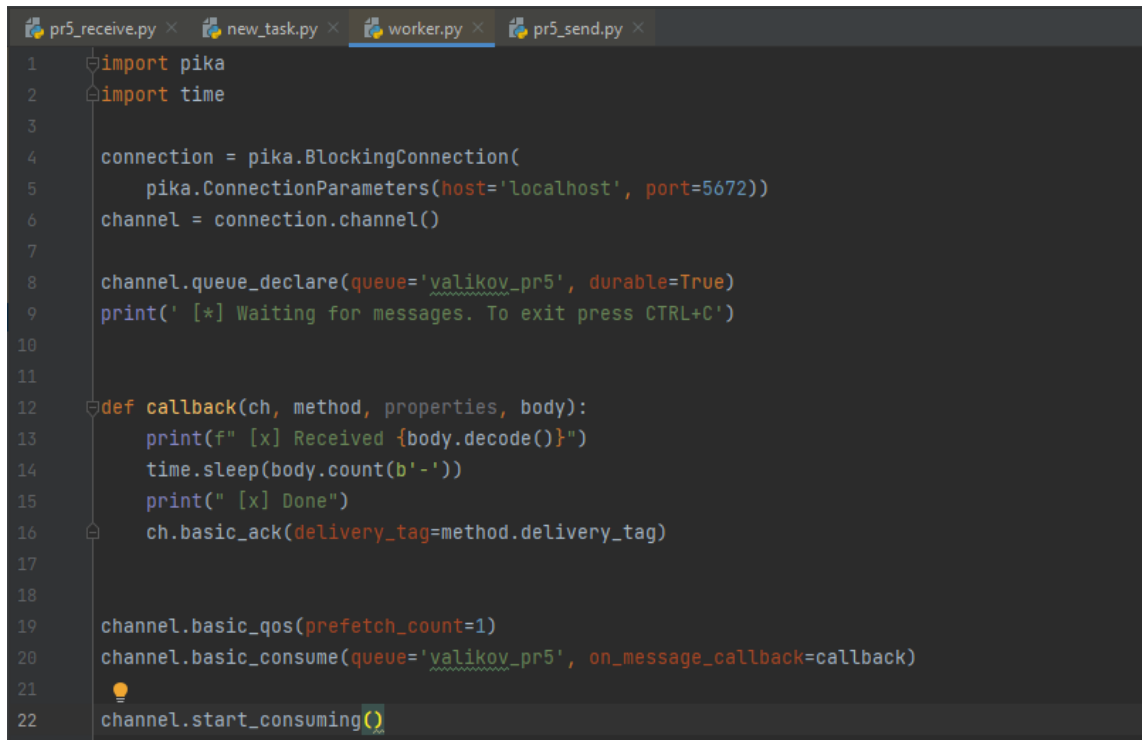
Page 1 of 0 - Filter:  ☐ Regex ?

... no queues ...

► Add a new queue

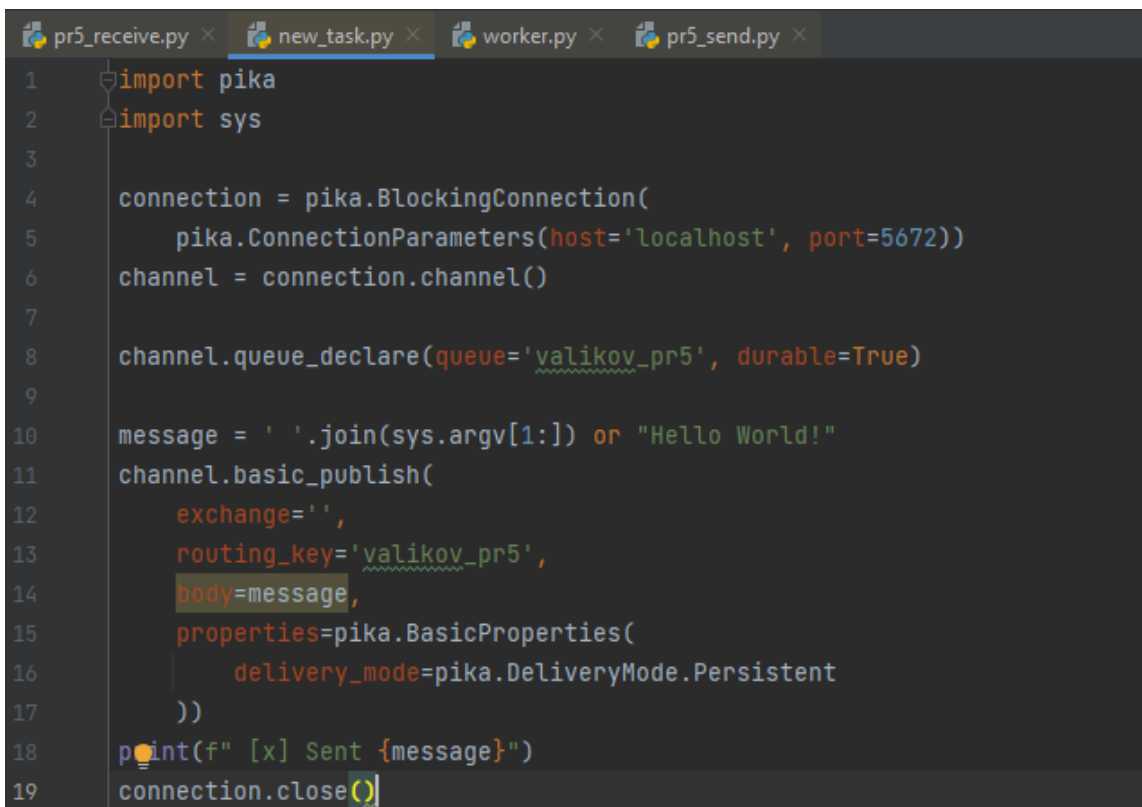
Рисунок 6. Очереди после остановки скрипта

- 2 Символ для обозначения времени сна: -. Тип обменника: topic. Сообщения должны храниться при выключении RabbitMQ.



```
1 import pika
2 import time
3
4 connection = pika.BlockingConnection(
5     pika.ConnectionParameters(host='localhost', port=5672))
6 channel = connection.channel()
7
8 channel.queue_declare(queue='valikoy-pr5', durable=True)
9 print(' [*] Waiting for messages. To exit press CTRL+C')
10
11
12 def callback(ch, method, properties, body):
13     print(f" [x] Received {body.decode()}")
14     time.sleep(body.count(b'-'))
15     print(" [x] Done")
16     ch.basic_ack(delivery_tag=method.delivery_tag)
17
18
19 channel.basic_qos(prefetch_count=1)
20 channel.basic_consume(queue='valikoy-pr5', on_message_callback=callback)
21
22 channel.start_consuming()
```

Рисунок 7. worker.py



```
1 import pika
2 import sys
3
4 connection = pika.BlockingConnection(
5     pika.ConnectionParameters(host='localhost', port=5672))
6 channel = connection.channel()
7
8 channel.queue_declare(queue='valikoy-pr5', durable=True)
9
10 message = ' '.join(sys.argv[1:]) or "Hello World!"
11 channel.basic_publish(
12     exchange='',
13     routing_key='valikoy-pr5',
14     body=message,
15     properties=pika.BasicProperties(
16         delivery_mode=pika.DeliveryMode.Persistent
17     ))
18 print(f" [x] Sent {message}")
19 connection.close()
```

Рисунок 8. new\_task.py

```
Windows PowerShell
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\worker.py
[*] Waiting for messages. To exit press CTRL+C
[x] Received Second Message
[x] Done
[x] Received Fourth Message
[x] Done

Windows PowerShell
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\worker.py
[*] Waiting for messages. To exit press CTRL+C
[x] Received First Message
[x] Done
[x] Received Third Message
[x] Done

Windows PowerShell
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\new_task.py First Message
[x] Sent First Message
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\new_task.py Second Message
[x] Sent Second Message
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\new_task.py Third Message
[x] Sent Third Message
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\new_task.py Fourth Message
[x] Sent Fourth Message
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики>
```

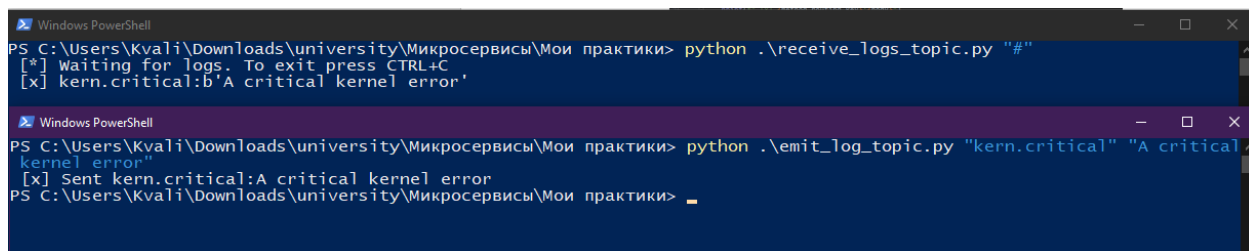
Рисунок 9. Результат работы

```
pr5_receive.py x emit_log_topic.py x receive_logs_topic.py x worker.py x pr5_send.py x
1 import pika
2 import sys
3
4 connection = pika.BlockingConnection(
5     pika.ConnectionParameters(host='localhost', port=5672))
6 channel = connection.channel()
7
8 channel.exchange_declare(exchange='topic_logs', exchange_type='topic')
9
10 routing_key = sys.argv[1] if len(sys.argv) > 2 else 'anonymous.info'
11 message = ' '.join(sys.argv[2:]) or 'Hello World!'
12 channel.basic_publish(
13     exchange='topic_logs', routing_key=routing_key, body=message)
14 print(f" [x] Sent {routing_key}:{message}")
15 connection.close()
```

Рисунок 10. emit\_log\_topic.py

```
pr5_receive.py x emit_log_topic.py x receive_logs_topic.py x worker.py x pr5_send.py x
1 import pika
2 import sys
3
4 connection = pika.BlockingConnection(
5     pika.ConnectionParameters(host='localhost', port=5672))
6 channel = connection.channel()
7
8 channel.exchange_declare(exchange='topic_logs', exchange_type='topic')
9
10 result = channel.queue_declare('', exclusive=True)
11 queue_name = result.method.queue
12
13 binding_keys = sys.argv[1:]
14 if not binding_keys:
15     sys.stderr.write("Usage: %s [binding_key]...\n" % sys.argv[0])
16     sys.exit(1)
17
18 for binding_key in binding_keys:
19     channel.queue_bind(
20         exchange='topic_logs', queue=queue_name, routing_key=binding_key)
21
22 print(' [*] Waiting for logs. To exit press CTRL+C')
23
24
25 def callback(ch, method, properties, body):
26     print(f" [x] {method.routing_key}:{body}")
27
28
29 channel.basic_consume(
30     queue=queue_name, on_message_callback=callback, auto_ack=True)
31
32 channel.start_consuming()
```

Рисунок 11. receive\_logs\_topic.py



The image shows two overlapping Windows PowerShell windows. The top window has a title bar with a blue icon and the text 'Windows PowerShell'. It contains the following text:  
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\receive\_logs\_topic.py "#"  
[\*] waiting for logs. To exit press CTRL+C  
[x] kern.critical:b'A critical kernel error'

The bottom window also has a title bar with a blue icon and the text 'Windows PowerShell'. It contains the following text:  
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики> python .\emit\_log\_topic.py "kern.critical" "A critical kernel error"  
[x] Sent kern.critical:A critical kernel error  
PS C:\Users\Kvali\Downloads\university\Микросервисы\Мои практики>

Рисунок 12. Результат работы